FORCEWORLD: A FORCE WORLD SIMULATOR WITH VISUAL AND FORCE FEEDBACK

Anonymous authors
Paper under double-blind review

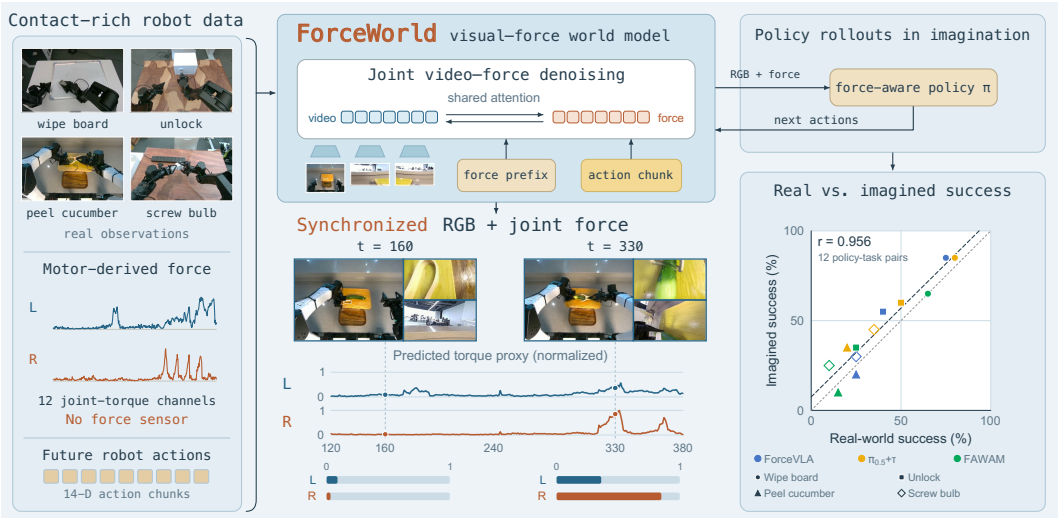


Figure 1: **ForceWorld jointly imagines visual and force observations for closed-loop policy evaluation.** Left: real observations and joint torques from four contact-rich tasks. Center: joint video–force predictions aligned with generated visual observations. Right: a force-aware policy acts on generated RGB and torque, yielding imagined success rates strongly correlated with real-world results across three policy families and four tasks. Anonymized project page with videos of all results: `https://anonymous.4open.science/w/ForceWorld-3FDC/`

ABSTRACT

Many existing world models focus on predicting future visual observations conditioned on robot actions. In contact-rich manipulation, however, distinct physical outcomes can appear visually similar. In such cases, force signals can provide key information about how the robot interacts with its environment. Therefore, we present ForceWorld, an action-conditioned world simulator that provides visual and force feedback to support closed-loop rollouts and evaluation of force-aware policies. Built on a pretrained video diffusion transformer, ForceWorld jointly denoises video and force representations to generate multi-view RGB observations and external joint-torque. To capture evolving contact conditions and their dependence on robot actions, we therefore augment framewise torque embeddings with a temporal encoder and directly modulate torque denoising with action inputs. Across four real-world contact-rich manipulation tasks, ForceWorld reduces high-load torque prediction RMSE by 26.6% compared with TACO, while also outperforming both evaluated video-prediction baselines. The predicted feedback enables closed-loop policy rollouts over minute-scale task horizons. This supports downstream policy evaluation, with success rates strongly correlated with those in real-world execution (Pearson $r = 0.956$).

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.

1 INTRODUCTION

World models predict the visual outcome of an action sequence (Yang et al., 2024; Zhu et al., 2024; NVIDIA, 2025), and are increasingly used as learned simulators for planning and policy evaluation (Zhu et al., 2024; Guo et al., 2025). In contact-rich manipulation, however, visual observations can be ambiguous about the physical interaction between the robot and its environment (Wang et al., 2026; Deng et al., 2026). For example, when screwing a bulb into a socket, smooth threading and jamming may be difficult to distinguish visually, while force feedback can help reveal differences in contact states. This motivates a world model that jointly predicts visual and force observations.

Recent work has extended video prediction with tactile images or latents (Higuera et al., 2026; Zheng et al., 2026; Huang et al., 2026) and hierarchical contact representations (Ma et al., 2026). TACO jointly generates video and fingertip force–torque sequences as local correction segments, which are subsequently labeled with actions for policy post-training (Liu et al., 2026). These advances broaden the contact information that world models can predict. This raises a further question: can jointly generated visual and force feedback sustain closed-loop policy execution and reflect real-world task performance?

To this end, we introduce ForceWorld, an action-conditioned world simulator that jointly predicts multi-view video and external joint torques for both arms. Given initial image and torque observations and a subsequent action chunk, the model predicts the feedback used by the policy to produce the next action chunk, enabling force-aware policy evaluation. Within a pretrained Wan video diffusion transformer (Wan Team, 2025), temporally aligned video and torque tokens interact through shared attention for joint denoising. For training, external joint-torque targets are obtained through sensorless estimation (Oh et al., 2026).

For closed-loop interaction, torque feedback needs to reflect evolving contact conditions and their dependence on robot actions. We augment framewise torque embeddings with a residual temporal encoder that adds temporal context while retaining a representation for each frame. To strengthen action conditioning in the torque branch, a dedicated action-modulation path directly modulates torque denoising with action inputs. With many near-zero free-motion targets in the training data, we use a load-weighted torque loss to place greater emphasis on high-load intervals.

Across four real-world contact-rich manipulation tasks, ForceWorld reduces torque prediction RMSE during high-load intervals by 26.6% compared with TACO and outperforms both evaluated video-prediction baselines. Adding torque prediction preserves action controllability and scene consistency in generated rollouts, while improving the physical plausibility of contact interactions. Closed-loop policy rollouts span minute-scale task horizons. Evaluation of three force-aware policy families shows a strong correlation between simulated and real-world success rates.

Our contributions are:

- We introduce ForceWorld, an action-conditioned world simulator that jointly denoises multi-view video and bilateral external joint torques, providing visual and force feedback for closed-loop execution of force-aware policies. It improves visual prediction quality and torque prediction accuracy in contact-rich manipulation.
- We develop torque-specific modeling and training mechanisms: TARE incorporates temporal context into framewise force embeddings; AMP directly modulates force tokens with actions; and PACE emphasizes contact steps in the torque loss.
- We demonstrate the use of ForceWorld as a closed-loop evaluation environment for force-aware policies. Across three policy families and four tasks, policy success rates in ForceWorld strongly correlate with real-world success rates (Pearson $r = 0.956$), with low policy-ranking error (task-averaged MMRV of 0.0125).

2 RELATED WORK

Action-conditioned world models. World models predict the consequences of actions so that an agent can plan or learn in imagination (Ha & Schmidhuber, 2018; Hafner et al., 2023). Action-conditioned video generators serve as learned simulators for manipulation (Yang et al., 2024; Zhu et al., 2024), while large-scale video pretraining supports both generative world models (Bruce et al.,

2024; NVIDIA, 2025) and latent predictive models for robotic planning (Assran et al., 2025). Ctrl-World further adapts pretrained video models for closed-loop policy execution (Guo et al., 2025). The same models increasingly serve to evaluate policies, with imagined success tracking real success across checkpoints (Li et al., 2025; 2026); dWorldEval also documents that a model finetuned on successful demonstrations tends to render failing actions as successes. Existing evaluation-oriented robot world models primarily expose generated visual observations during imagined policy execution. ForceWorld extends this setting by jointly predicting the external joint-torque feedback consumed by force-aware policies, enabling closed-loop evaluation under both visual and torque-dependent observations.

Contact in world models. A growing line of work predicts contact alongside video. Most of it predicts a tactile image or its latent (Higuera et al., 2026; Zheng et al., 2026; Huang et al., 2026). Others predict discrete contact states such as contact, deformation, and slip (Ma et al., 2026), or a numeric force–torque sequence co-denoised with the frames (Liu et al., 2026). Tactile world action models predict such futures together with the actions themselves (Lou et al., 2026; Tian et al., 2026). The predicted contact serves planning (Higuera et al., 2026; Ma et al., 2026), data generation for downstream policies (Huang et al., 2026), or corrective segments for post-training (Liu et al., 2026). Our setting differs in both the force representation and its role in the rollout. Rather than relying on a dedicated tactile or force-torque sensor at the end effector, we model bilateral external joint-torque proxies derived from motor currents. More importantly, the predicted torque is returned to force-aware policies as part of their observation stream during closed-loop imagined execution, enabling policy evaluation under the same visual-and-torque feedback structure used during real interaction.

3 METHOD

We introduce ForceWorld, an action-conditioned world simulator that jointly predicts future visual observations and external joint-torque proxies, enabling closed-loop evaluation of force-aware policies through predicted visual and force feedback. Torque supervision is derived from motor currents without dedicated force sensors (Section 3.1). We encode the torque sequence as force tokens and jointly denoise them with video latents in a pretrained Wan video diffusion transformer (Section 3.2). Action modulation is applied directly to the force tokens, allowing force prediction to use actions without relying solely on action information passed through the visual tokens. We assign higher force-loss weights to contact steps identified from ground-truth torque magnitude, counteracting the predominance of near-zero free-motion steps (Section 3.3). Figure 2 shows the architecture; implementation details and hyperparameters are provided in Appendix A.

3.1 FORCE SUPERVISION WITHOUT FORCE SENSORS

To provide force supervision for our world model without dedicated force sensors, we estimate external joint torque from motor currents. Motor torque reflects both the effort required to move the arm itself and external torque. Following Neural External Torque Estimation (NEXT) (Oh et al., 2026), we train an LSTM on contact-free trajectories to predict the torque required for the arm's own motion from a local history of joint positions, velocities and tracking errors. During manipulation, the robot continues to record motor currents, which reflect the motor effort under the actual load. To generate supervision offline, we convert these measured currents into motor torque and subtract the free-space torque predicted by NEXT from the corresponding joint-state history:

$$\widehat{\boldsymbol{\tau}}^{\mathrm{ext}}_{a,t} = \boldsymbol{\tau}^{m}_{a,t} - \widehat{\boldsymbol{\tau}}^{f}_{a,t}, \qquad \boldsymbol{\tau}^{m}_{a,t} = \mathbf{K}_a \mathbf{I}^{m}_{a,t}. \tag{1}$$

Here, $\mathbf{I}^{m}_{a,t}$ is the motor-current vector for arm a at time t, $\mathbf{K}_a$ is a diagonal matrix of per-joint scaling factors, and $\widehat{\boldsymbol{\tau}}^{f}_{a,t}$ is the predicted free-space torque. We use the residual as an external joint-torque proxy, concatenating six joint values from each arm into the 12-D supervision joint state $\boldsymbol{\tau}_t$. Appendix A details the estimator inputs and history window.

3.2 JOINT VIDEO–FORCE CO-DENOISING

We index a prediction window by $t = 1, \ldots, T$, with $\mathbf{x}_1$ the current visual observation, $\widetilde{\boldsymbol{\tau}}_1$ its normalized force proxy, and $\mathbf{a}_{1:T}$ the aligned action sequence. Each action is a 14-D vector that concatenates the 7-D end-effector pose of each arm.

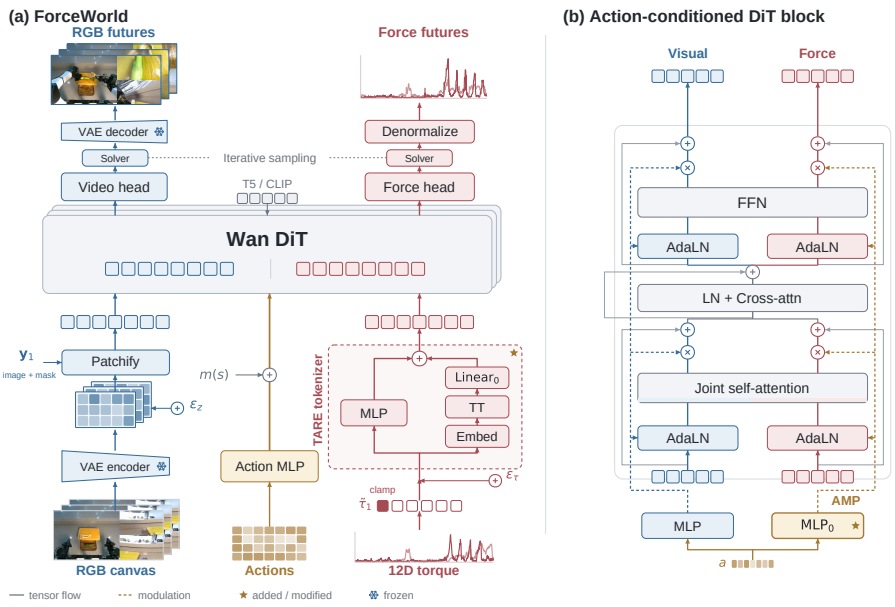


Figure 2: **ForceWorld architecture.** (a) Video and force tokens are jointly denoised in a shared Wan DiT, conditioned on the current observation and actions. TARE adds temporal context to force embeddings; the observed force remains a clean anchor. (b) Video and force tokens share attention and FFN weights, with separate action modulation; AMP directly conditions force tokens on actions. Snowflakes denote frozen modules, stars mark added or modified components.

Multimodal representation. We encode the multi-view video into latents $\mathbf{z}$ using the pretrained Wan VAE and apply the backbone's patch embedding to obtain N_v visual tokens of width d. Our force tokenizer maps the normalized torque sequence $\widetilde{\boldsymbol{\tau}}_{1:T} \in \mathbb{R}^{T\times 12}$ into T tokens of the same width, retaining one force token per sampled video frame. At each denoising step, we concatenate the visual and force tokens along the token dimension to form a sequence of shape $(N_v + T) \times d$, processed by the shared DiT. To establish a common temporal coordinate system, we map each force time step to a video-latent time index and reuse the corresponding temporal RoPE. Force tokens have no image-plane location, so their spatial rotary transformations are set to identity. The temporal mapping and video tokenization details are provided in Appendix A.

Shared denoising. We perturb the video latents and normalized torque sequence with independent Gaussian noise at a shared diffusion time s drawn from the backbone's training schedule:

$$\mathbf{z}^s = (1-s)\mathbf{z} + s\boldsymbol{\epsilon}_z, \qquad \widetilde{\boldsymbol{\tau}}^s = (1-s)\widetilde{\boldsymbol{\tau}} + s\boldsymbol{\epsilon}_\tau. \tag{2}$$

The force at the observed frame is kept clean as a conditioning anchor (Section 3.3). Conditioned on the current observation and actions, the shared DiT processes the resulting video and force tokens and predicts both flow fields:

$$(\widehat{\mathbf{v}}^z, \widehat{\mathbf{v}}^\tau) = \mathrm{DiT}_\theta([\,\mathcal{P}(\mathbf{z}^s; \mathbf{y}_1)\;;\; \mathbf{f}^s_{1:T}\,], s, \mathbf{a}_{1:T}, \mathbf{e}_{\text{img}}, \mathbf{e}_{\text{text}})\,. \tag{3}$$

Here, $\mathcal{P}$ embeds the noisy video latents together with the observed-image latent and mask condition $\mathbf{y}_1$, while $\mathbf{f}^s_{1:T}$ denotes the force tokens constructed from the perturbed torque sequence with its clean anchor. The image and text embeddings $\mathbf{e}_{\text{img}}$ and $\mathbf{e}_{\text{text}}$ provide additional context through cross-attention. Shared self-attention allows the evolving video and force representations to inform each other at every denoising step. Separate output heads predict the video and force flows, with targets $\boldsymbol{\epsilon}_z - \mathbf{z}$ and $\boldsymbol{\epsilon}_\tau - \widetilde{\boldsymbol{\tau}}$, respectively. The full joint training objective is given in Appendix A.

Temporal Additive Residual Embedding (TARE). Force trajectories encode contact through temporal patterns such as load onset and sustained loading. A framewise tokenizer leaves these patterns entirely to the shared video backbone. We therefore add a force-specific temporal transformer

(TT) branch that incorporates sequence context before the DiT, while retaining a direct framewise embedding through a residual connection:

$$\mathbf{f}_t^s = \phi(\widetilde{\boldsymbol{\tau}}_t^s) + g_0\big(\mathrm{TT}(\psi(\widetilde{\boldsymbol{\tau}}_{1:T}^s) + \mathbf{e}_{\mathrm{pos}} + \mathbf{e}_{\mathrm{time}}(s))_t\big)\,. \tag{4}$$

Here, t indexes sampled trajectory frames and s denotes diffusion time. The MLP ϕ maps each 12-D torque vector directly to the DiT width d. In the temporal branch, an independent MLP ψ embeds each vector at a smaller width, and TT exchanges information across the sequence using position and diffusion-time embeddings. The affine projection g_0 maps each output back to width d for addition to the framewise embedding. Its weights and bias are zero-initialized, so TARE starts as a framewise tokenizer and learns temporal corrections during training. Section 4.3 compares it with framewise tokenizers.

Action Modulation Path (AMP). Our action conditioner groups C consecutive actions and maps each chunk to an additive timestep embedding and six AdaLN modulation vectors for the corresponding visual tokens. We extend it with a separate force-modulation head, so force tokens can access actions directly as well as through shared attention. Force token t receives the six vectors for chunk $\pi(t)$, using the temporal alignment defined in the appendix. These vectors add to its base timestep modulation; the additive timestep embedding remains on the visual path. The force-modulation head's final projection is zero-initialized, preserving the model without direct force-action modulation at initialization. To discourage reliance on this direct action branch, modulation dropout zeros all its vectors for a sample with probability p during training. The branch is active at inference. Component ablations in Section 4.3 assess the contribution of the proposed mechanisms.

3.3 Training Objective

We jointly train the video and force streams with flow-matching losses, using the perturbations and targets in Section 3.2. The initial force is treated as an observed condition, and high-load steps receive additional weight in the force objective. The full training objective and optimization details are provided in Appendix A.

Observed-force anchor. For the window indexed in Section 3.2, we condition on the normalized torque proxy $\widetilde{\boldsymbol{\tau}}_1$ aligned with the current image. During training, we replace its noised value with the clean label before tokenization and exclude this entry from the force loss. The supervised force indices are therefore $\mathcal{F} = \{2, \ldots, T\}$. During sampling, the same entry is initialized to the supplied anchor and restored after every solver step. The temporal tokenizer retains the shared noise-level embedding; the anchor's base DiT timestep modulation uses zero (Appendix A).

Phase-Aware Contact Emphasis (PACE). To prioritize force prediction during high-load portions of a trajectory, we reweight time steps using clean normalized torque labels. For arm $a \in \{\mathrm{L}, \mathrm{R}\}$, let $\widetilde{\boldsymbol{\tau}}_{a,t}$ contain its six joint values, and let κ_a be its frozen training-set threshold. We define $c_t = \mathbb{1}[\exists a : \|\widetilde{\boldsymbol{\tau}}_{a,t}\|_2 > \kappa_a]$ and $w_t = 1 + \lambda_c c_t$. The same weight applies to all 12 joints at time t when either arm exceeds its threshold. This defines a high-load proxy for contact-related phases, rather than an independently observed contact state. The weights depend only on training labels and require no contact detector at inference. Let $\ell_t = \|\widehat{\mathbf{v}}_t^{\tau} - \mathbf{v}_t^{\tau}\|_2^2/12$ be the mean squared flow error across joints. The force loss is

$$\mathcal{L}_{\mathrm{force}} = \frac{\sum_{t\in\mathcal{F}} w_t \ell_t}{\sum_{t\in\mathcal{F}} w_t}. \tag{5}$$

At $\lambda_c = 0$ this reduces to uniform mean squared error over future force entries. Positive λ_c increases the relative weight of high-load steps while retaining supervision on lower-load steps. Normalization removes a common multiplicative factor in the weights; consequently, weighting changes the objective only when a supervised window contains both strata. Table 2 evaluates the trade-off between their prediction errors.

4 Experiments

Our experiments examine whether jointly predicting video and external joint torque enables Force-World to model contact-rich interactions and evaluate force-aware policies. We first assess the sensorless torque labels that provide supervision (Section 4.2). We then evaluate video and torque

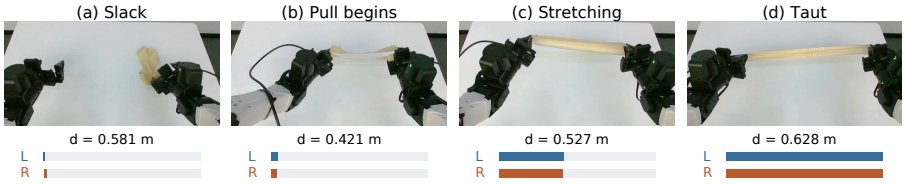


Figure 3: Sensorless force estimation during elastic-band stretching. Four frames show gripper distance d and estimated external torque magnitudes for both arms (L/R) on a shared linear scale. Both readings rise similarly as the band becomes taut.

prediction, isolate the contributions of the model's design choices(Section 4.3), and test whether joint torque prediction helps generated videos respect contact constraints(Section 4.4). Finally, we evaluate force-aware policies in closed-loop rollout and assess how well imagined success rates and policy rankings align with real-world performance(Section 4.5).

4.1 Experimental Setup

Data and models. We train a multi-task world model on four bimanual tasks: Wipe board, Unlock, Peel cucumber, and Screw bulb, using synchronized RGB images, actions, and torque. At 10 Hz, each prediction chunk contains one observed frame and 48 future frames, spanning 4.8 s; longer rollouts are generated autoregressively. The dataset contains equal numbers of successful and failed teleoperated trajectories and is split by trajectory into training, validation, and test sets. Training excludes rollouts from the evaluated policies (Appendix D). Unless otherwise specified, comparisons use matched training data and training and sampling budgets, with full configurations provided in Appendix B.1.

Evaluation setup. We evaluate video and torque prediction on held-out trajectories, conditioned on initial observations and recorded actions. For policy evaluation, we deploy the same policy checkpoints on the real robot and in the world model. In both environments, each policy query predicts a 50-action chunk, with actions executed at 10 Hz and periodic replanning. During imagined execution, the policy receives generated RGB and torque feedback at the end of the executed action segment. Forward kinematics converts the policy actions to end-effector poses for world-model action conditioning. We compare success rates and policy rankings across the two environments. Detailed protocols appear in the corresponding experiment sections and Appendices B.1 and D.

Evaluation metrics. We evaluate video prediction using PSNR, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018), and torque prediction using RMSE. For phase-specific evaluation, a timestep is classified as high-load when either arm's normalized reference torque-label vector has an ℓ_2 norm above its fixed, training-calibrated threshold. For policy evaluation, Pearson correlation and MMRV (Li et al., 2025) measure success-rate correlation and ranking consistency between real and imagined rollouts. Additional torque diagnostics and detailed protocols appear in Appendix B.2.

4.2 Experiment 1: Validation of Sensorless Torque Labels

Most robot arms carry no force sensor, so we take the force signal from what every actuated arm already logs, following NEXT (Oh et al., 2026). Whether the force obtained this way is valid has to be checked, and we check it in the two settings where the answer is known in advance. On contact-free trajectories held out from the labeler's training data, the estimated external-torque residuals remain near-zero. When the two arms pull an elastic band held between them, the force follows the stretch of the band: it is near zero while the band is slack and rises monotonically with the gripper distance once the band is taut, as is shown in Figure 3. The averaged Spearman's rank correlation coefficient over arms and pulls is $\rho = 0.97$.

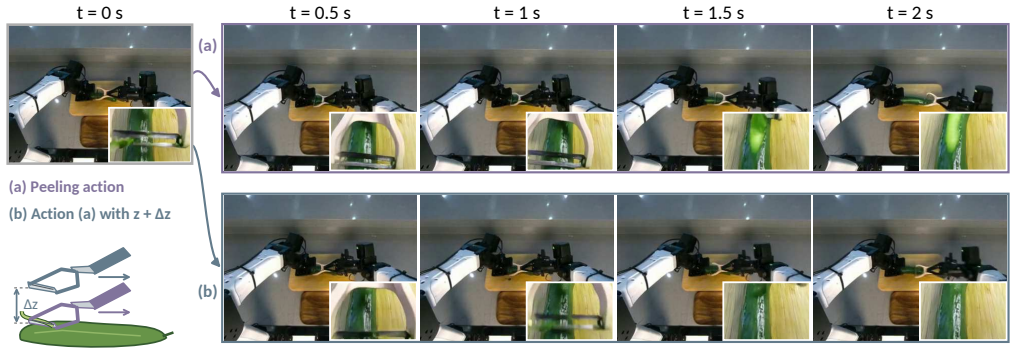


Figure 4: **Action-conditioned peeling predictions.** Starting from the same initial observations, the model predicts (a) peel removal under a peeling action and (b) no peel removal when the end-effector height in the action sequence is increased, causing the peeler to pass above the cucumber. Insets show enlarged left-wrist views of the interaction.

Table 1: Video and torque prediction on four tasks. Evaluation details are in Appendix B.2.

	RGB			Force RMSE by phase		Contact	
Model	PSNR↑	SSIM↑	LPIPS↓	Free↓	Contact↓	AUROC↑	Onset err. (fr.)↓
Ctrl-World (Guo et al., 2025)	13.44	0.535	0.508	–	–	–	–
TACO (Liu et al., 2026)	14.33	0.550	0.453	0.089	0.364	0.592	2.16
ForceWorld	**17.16**	**0.633**	**0.341**	**0.065**	**0.267**	**0.846**	**1.89**

Table 2: Component ablations on Peel cucumber. Rows with + add components cumulatively. Brackets denote 95% bootstrap intervals over clips. Protocol details are in Appendix B.2.

Variant	Free RMSE↓	Contact RMSE↓	Contact r ↑	AUROC↑	Canvas LPIPS↓
Framewise MLP + head	0.132 [0.124, 0.141]	0.317 [0.296, 0.341]	0.318 [0.289, 0.345]	0.774 [0.748, 0.800]	0.383 [0.374, 0.391]
TARE	0.132 [0.125, 0.140]	0.291 [0.272, 0.310]	0.380 [0.349, 0.408]	0.839 [0.817, 0.860]	0.379 [0.372, 0.386]
+ AMP	0.106 [0.100, 0.112]	0.288 [0.267, 0.311]	0.390 [0.358, 0.421]	0.846 [0.823, 0.868]	0.355 [0.348, 0.363]
+ PACE (full)	0.109 [0.102, 0.116]	**0.273** [0.252, 0.296]	**0.422** [0.389, 0.453]	**0.846** [0.821, 0.871]	**0.337** [0.328, 0.345]

4.3 Experiment 2: Video and Torque Prediction

Video quality and torque prediction accuracy. We evaluate video and torque prediction against TACO (Liu et al., 2026) and the video-only Ctrl-World (Guo et al., 2025). For video prediction, Table 1 shows that ForceWorld achieves higher PSNR and SSIM and lower LPIPS than both evaluated video baselines. For torque prediction, ForceWorld achieves lower torque-proxy RMSE than TACO in both free-motion (0.065 versus 0.089) and high-load contact phases (0.267 versus 0.364). A matched comparison on Peel cucumber Task further evaluates video quality with and without the torque prediction stream in Appendix B.3 and Table 4.

Action controllability. To examine whether actions can control the predicted interaction, we hold the initial observations fixed on Peel cucumber Task and vary the end-effector height in the action sequence. Figure 4 shows that raising the peeler changes the predicted outcome from peel removal to passing above the cucumber without contact.

Ablation of model components. We investigate whether temporal representation, direct action conditioning, and contact-focused supervision address different sources of prediction error. On Peel cucumber, Table 2 reports the complete torque and video metrics for the corresponding variants. TARE improves contact-phase prediction, whereas AMP mainly benefits free-motion accuracy and video quality. PACE further improves contact-phase error and correlation, with a small increase in free-motion error. This pattern suggests that the components address different aspects of the prediction problem. The full model achieves the lowest contact-phase RMSE and video LPIPS among the tested variants.

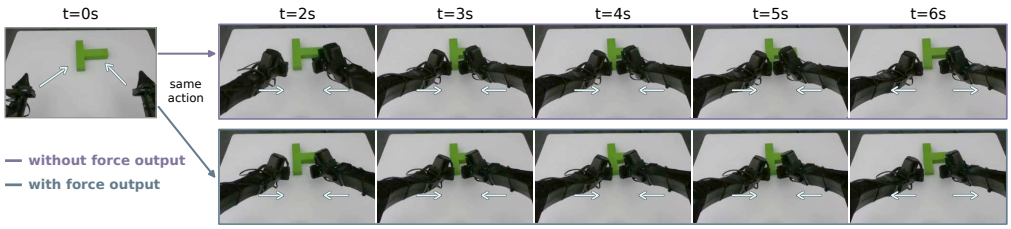


Figure 5: **Rollouts with and without torque prediction.** Given the same initial visual observations and action sequence on Push-T, the visual-only version (top) predicts the gripper passing through the rigid block, failing to respect the contact constraint. In contrast, ForceWorld (bottom) with force output captures contact-constrained motion: the gripper is blocked by the rigid object rather than passing through it. Arrows indicate actions.

Table 3: **Visual prediction on Push-T.** Comparison of visual-only and joint video–torque models in video quality and apparent penetration rate.

Model	PSNR↑	SSIM↑	LPIPS↓	Penetration↓
Visual-only	13.16	0.658	0.334	0.65
ForceWorld	**13.97**	**0.672**	**0.313**	**0.05**

4.4 Experiment 3: Effect of Joint Torque Prediction on Visual Contact Behavior

We study Push-T, where contact with a rigid block provides a clear visual test of predicted robot–object interactions. A gripper passing through the block produces observable interpenetration, making this task suitable for assessing whether joint torque prediction improves visual contact behavior.

Experimental setup. We compare ForceWorld with a visual-only counterpart using the same training data, action inputs, and prediction horizons. Both models start from pre-contact observations and predict future frames conditioned on recorded actions. We evaluate video quality against real trajectories and use manual visual assessment to measure apparent penetration: whether the predicted gripper passes into the block. See Appendix C for assessment details.

Results. Table 3 shows that ForceWorld outperforms the visual-only model across all three video-quality metrics, while reducing apparent penetration from 65% to 5%. Figure 5 illustrates the contact behavior under the same action sequence: the visual-only model predicts the gripper passing through the block, whereas ForceWorld better preserves the contact constraint.

4.5 Experiment 4: World-Model Evaluation of Force-Aware Policies

We examine whether ForceWorld can evaluate force-aware policies through closed-loop interaction. A comparison of policies with and without torque input on Screw bulb motivates the need for torque feedback in the evaluation environment. We then compare real and imagined success rates at the latest checkpoint of each policy family.

Why policy evaluation requires torque feedback. Force-aware policies use torque feedback to guide contact-rich manipulation. Evaluating these policies in a world simulator therefore requires reproducing both their visual and torque observations. Figure 6 illustrates this need on Screw bulb. Starting from the same scene in the same world model, the policy receiving predicted torque leaves the twisting pose by 30 s, whereas the policy without torque input continues twisting through 42 s. The two policies differ in whether force feedback is available in the evaluation loop. This example motivates a world model that supplies torque alongside images, allowing force-aware policies to be evaluated with the feedback they use during execution.

Policies and evaluation. We evaluate ForceVLA (Yu et al., 2025), $\pi_{0.5}{+}\tau_{\text{ext}}$, and FAWAM (He et al., 2026) on Wipe board, Unlock, Peel cucumber, and Screw bulb, with 20 real and 20 imagined rollouts per policy–task pair. During imagined rollouts, policies receive generated images and

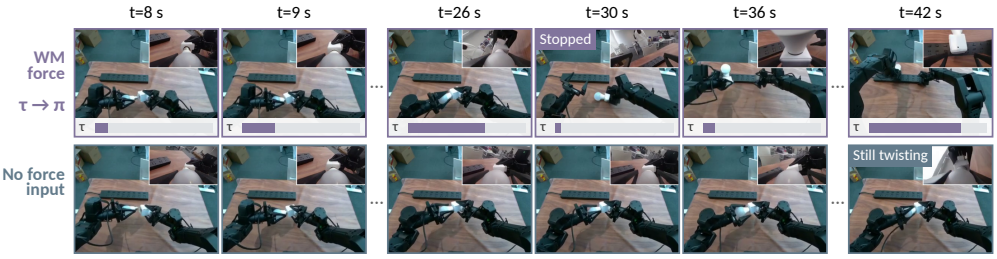


Figure 6: **Bulb screwing with and without torque feedback.** From the same scene, the policy with torque input (top) leaves the twisting pose while the policy without torque input (bottom) continues twisting. Bars show predicted torque magnitude on a shared scale; insets show wrist views.

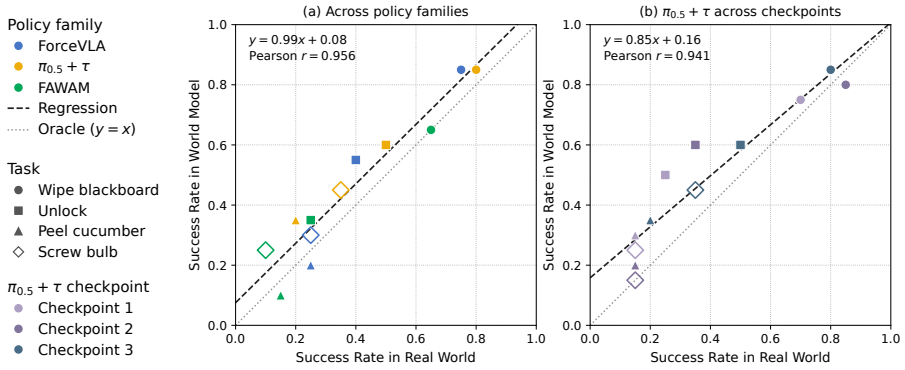


Figure 7: **Real versus imagined policy success with predicted torque feedback.** **(a)** Latest-checkpoint results across three policy families and four tasks; colors denote policy families. **(b)** $\pi_{0.5}+\tau_{\text{ext}}$ at three checkpoints across the same four tasks; colors denote checkpoints. Each panel fits its own unweighted least-squares line over 12 points (dashed).

predicted torque as feedback. Task success is manually annotated in both environments. We report Pearson correlation across the 12 policy–task pairs and MMRV computed within each task and averaged over the four tasks. Checkpoint details, success counts, and metric definitions are provided in Appendix D.

Comparison with real-world performance. Figure 7(a) compares real and imagined success rates across three policy families and four tasks. Across the 12 policy–task pairs, imagined evaluation with predicted torque feedback achieves a Pearson correlation of $r = 0.956$ with real-world success rates and a task-averaged MMRV of 0.0125. These results indicate strong success-rate correlation and low within-task ranking error for the evaluated policies. Figure 7(b) additionally compares three checkpoints of $\pi_{0.5}+\tau_{\text{ext}}$ across the four tasks, yielding a Pearson correlation of $r = 0.941$ between real and imagined success rates. Appendix D reports the success counts and metric definitions, and Appendix D.2 further examines the role of predicted torque feedback in policy evaluation.

5 Conclusion

We introduce ForceWorld, an action-conditioned world simulator that jointly forecasts multi-view RGB and bilateral external joint-torque proxies. Built on a pretrained Wan visual backbone, ForceWorld couples video latents and temporal force tokens in synchronized rollouts and feeds predicted torque observations back to policies for closed-loop evaluation with force feedback. On the evaluated tasks, ForceWorld achieves better visual prediction quality than Ctrl-World and TACO and lower torque-proxy prediction errors than TACO. Across three policy families on four tasks, imagined success rates correlate strongly with real success rates ($r = 0.956$), while a task-averaged MMRV of 0.0125 indicates broadly consistent within-task policy rankings.

AI USE STATEMENT

In preparing this manuscript, we used generative AI tools to support writing and language editing by refining wording, improving grammar and clarity, and enhancing the organization of the text. Every AI-assisted revision incorporated into the manuscript was reviewed and verified by the authors. The authors assume full responsibility for the accuracy and integrity of the final content, including its scientific claims, experimental results, and references.

REFERENCES

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, et al. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025. URL https://arxiv.org/abs/2506.09985.

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, et al. Genie: Generative interactive environments. In *International Conference on Machine Learning*, 2024. URL https://arxiv.org/abs/2402.15391.

Haoyuan Deng, Haichao Liu, Wenkai Guo, Yuan Ling, Zaijia Yang, Yuanjiang Xue, Haosheng Sun, Liangzi Wang, and Ziwei Wang. Facet-0: A robotic foundation model for contact-rich precise manipulation. *arXiv preprint arXiv:2609.01596*, 2026. URL https://arxiv.org/abs/2609.01596.

Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-World: A controllable generative world model for robot manipulation. *arXiv preprint arXiv:2510.10125*, 2025. URL https://arxiv.org/abs/2510.10125.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018. URL https://arxiv.org/abs/1803.10122.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023. URL https://arxiv.org/abs/2301.04104.

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. URL https://arxiv.org/abs/2606.08555.

Carolina Higuera, Sergio Arnaud, Byron Boots, Mustafa Mukadam, Francois Robert Hogan, and Franziska Meier. Visuo-tactile world models. *arXiv preprint arXiv:2602.06001*, 2026. URL https://arxiv.org/abs/2602.06001.

Yunao Huang, Shiyu Sang, Haotao Lu, Suting Ni, Shijie Wu, Ziyang Guo, Ye Shi, and Jingya Wang. ViTacWorld: Scaling visuo-tactile world models for contact-rich robot manipulation. *arXiv preprint arXiv:2607.22530*, 2026. URL https://arxiv.org/abs/2607.22530.

Yaxuan Li, Yichen Zhu, Junjie Wen, Chaomin Shen, and Yi Xu. WorldEval: World model as real-world robot policies evaluator. *arXiv preprint arXiv:2505.19017*, 2025. URL https://arxiv.org/abs/2505.19017.

Yaxuan Li, Zhongyi Zhou, Yefei Chen, Yaokai Xue, and Yichen Zhu. dWorldEval: Scalable robotic policy evaluation via discrete diffusion world model. *arXiv preprint arXiv:2604.22152*, 2026. URL https://arxiv.org/abs/2604.22152.

Shengbang Liu, Yueru Jia, Yuyang Yan, Jiaming Liu, Xinran Zhang, Qiuxuan Feng, Yandong Guo, Shiji Zhou, Boxin Shi, and Shanghang Zhang. TACO: Tactile world model as a self-corrector for scalable VLA post-training. *arXiv preprint arXiv:2607.02840*, 2026. URL https://arxiv.org/abs/2607.02840.

Yunfan Lou, Yifan Ye, Yankai Fu, Jun Cen, Xiaowei Chi, Yaoxu Lyu, Peidong Jia, Sirui Han, Zhihe Lu, and Shanghang Zhang. Dream-Tac: A unified tactile world action model for contact-rich robot manipulation. *arXiv preprint arXiv:2606.08737*, 2026. URL https://arxiv.org/abs/2606.08737.

Wenxuan Ma, Chaofan Zhang, Chao Xue, Yinghao Cai, Guocai Yao, Shaowei Cui, and Shuo Wang. FeelWorld: Visuo-tactile world model for hierarchical contact prediction and planning. *arXiv preprint arXiv:2607.24267*, 2026. URL https://arxiv.org/abs/2607.24267.

NVIDIA. Cosmos world foundation model platform for physical AI. *arXiv preprint arXiv:2501.03575*, 2025. URL https://arxiv.org/abs/2501.03575.

Steven Oh, Jason Jingzhou Liu, Tony Tao, Philip Han, Kenneth Shaw, Satoshi Funabashi, Ruslan Salakhutdinov, and Deepak Pathak. FACTR 2: Learning external force sensing for commodity robot arms improves policy learning. *arXiv preprint arXiv:2606.12406*, 2026. URL https://arxiv.org/abs/2606.12406.

Shuai Tian, Yupeng Zheng, Yuhang Zheng, Songen Gu, Yujie Zang, Yuxing Qin, Weize Li, Haoran Li, Wenchao Ding, and Dongbin Zhao. VT-WAM: Visual-tactile world action model for contact-rich manipulation. *arXiv preprint arXiv:2607.02503*, 2026. URL https://arxiv.org/abs/2607.02503.

Wan Team. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. URL https://arxiv.org/abs/2503.20314.

Mingxin Wang, Zhirun Yue, Renhao Lu, Yizhe Li, Zihan Wang, Guoping Pan, Kangkang Dong, Jun Cheng, Yi Cheng, and Houde Liu. PhaForce: Phase-scheduled visual-force policy learning with slow planning and fast correction for contact-rich manipulation. *arXiv preprint arXiv:2603.08342*, 2026. URL https://arxiv.org/abs/2603.08342.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

Sherry Yang, Yilun Du, Seyed Kamyar Seyed Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.06114.

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. *arXiv preprint arXiv:2505.22159*, 2025. URL https://arxiv.org/abs/2505.22159.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018.

Yuhang Zheng, Songen Gu, Weize Li, Yupeng Zheng, Yujie Zang, Shuai Tian, Xiang Li, Ruihai Wu, Ce Hao, Chen Gao, Si Liu, Haoran Li, Yilun Chen, Shuicheng Yan, and Wenchao Ding. OmniVTA: Visuo-tactile world modeling for contact-rich robotic manipulation. *arXiv preprint arXiv:2603.19201*, 2026. URL https://arxiv.org/abs/2603.19201.

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. IRASim: Learning interactive real-robot action simulators. *arXiv preprint arXiv:2406.14540*, 2024. URL https://arxiv.org/abs/2406.14540.

A Implementation and Training Details

Sensorless torque-label generation. The NEXT checkpoint used for label generation takes a 25-step history at 50 Hz (0.5 s) for each arm. Its 18-dimensional input at each step concatenates six joint positions, six joint velocities, and six position-tracking errors between commanded and measured joints. A two-layer LSTM with hidden size 128 and dropout 0.1 feeds a two-layer MLP head with hidden size 256, which predicts the six-dimensional free-space current proxy at the final step of the window. The left and right arms share this estimator, and the recurrent state is reset for every sliding window.

Multimodal input representation. Three RGB views are composed into a 288×640 canvas before video preprocessing and VAE encoding. The force stream concatenates six joint-torque proxy values from each arm into a 12D sequence, while the action stream uses the 14D end-pose representation defined in Section 3.2. Each training window contains $T = 49$ frames, which the Wan VAE temporally compresses into $L = 13$ latent frames. We retain every fifth timestep from the raw logs. Each force dimension is linearly scaled to $[-1, 1]$ using the 1st and 99th percentiles computed on the training split.

Backbone and conditioning. We initialize the DiT from the released 14B image-to-video Wan 2.1 checkpoint (Wan Team, 2025) ($d = 5120$) and fully finetune it together with the added force and action modules. The action conditioner groups $C = 4$ actions per latent frame using a hidden width of 1024, and produces an additive timestep embedding and six-way AdaLN modulation at the video-latent temporal resolution. A separate zero-initialized AMP head applies the same form of action modulation directly to the force tokens. In Equation 4, the framewise branch ϕ maps $12 \to 512 \to d$ with a SiLU activation. The temporal residual branch first maps the input through $\psi : 12 \to 512$, applies a 2-layer, 8-head transformer of width 512, and projects the result to d through the zero-initialized g_0. A per-token MLP with one hidden layer of width 512 predicts the 12D force output.

Token geometry and temporal alignment. Each 49-frame 288×640 window is encoded into a $13 \times 36 \times 80$ VAE latent grid with 16 channels. The DiT patch embedding of size $(1, 2, 2)$ produces $13 \times 18 \times 40 = 9360$ visual tokens of width $d = 5120$. The three camera views are encoded jointly as regions of one tiled canvas rather than by separate VAEs. We append one force token per input frame, giving 49 additional tokens. Force token t is assigned the latent-time index $\pi(t) = \mathrm{round}((t-1)(L-1)/(T-1)) + 1$, and uses the corresponding temporal RoPE with identity spatial components. The same mapping aligns AMP modulation with the force tokens. This interpolation-based index is an alignment convention. After frame subsampling, the 49 actions are divided into $L = 13$ chunks of $C = 4$; the final action is repeated three times to complete the last chunk.

Noise schedule and anchor embeddings. Training uniformly samples one of 1000 discrete scheduler indices. Before the Wan shift, the corresponding noise levels are $u_i = 1 - i/1000$, $i = 0, \ldots, 999$; applying the default shift of 5 gives $s_i = 5u_i/(1 + 4u_i)$, and the network receives timestep $1000 s_i$. The resulting noise levels are therefore discrete and nonuniform on $[0, 1]$. Both flow-matching losses are multiplied by the scheduler weight $\omega(s_i) \propto \exp[-2(s_i - 1/2)^2] - \min_j \exp[-2(s_j - 1/2)^2]$, normalized to have unit mean over the 1000 entries. For the observed-force anchor, we first construct the noised force sequence and then replace its first entry with the clean value before tokenization; this entry is excluded from the force loss. The temporal tokenizer still uses the shared diffusion-time embedding, while the DiT applies timestep-zero base modulation to the anchor token and timestep $1000 s_i$ to the remaining force tokens.

Loss reduction and supervised regions. We optimize the joint video and force flow-matching objective

$$\mathcal{L} = \mathbb{E}[\omega(s)\,(\mathcal{L}_{\mathrm{video}} + \lambda_\tau \mathcal{L}_{\mathrm{force}})]\,, \tag{6}$$

where $\mathcal{L}_{\mathrm{video}}$ is the mean squared video-flow error over all supervised latent entries, $\mathcal{L}_{\mathrm{force}}$ is defined in Equation 5, and $\omega(s)$ is the scheduler's timestep loss weight described above. The expectation is over training windows, the shared noise level, and independent Gaussian noises for the two modalities. The two losses are separately mean-reduced, so λ_τ controls their relative weighting.

In the co-denoising path, squared force-flow errors are averaged over the 12 joint dimensions and all supervised timesteps. A timestep is assigned the high-load weight $1 + \lambda_c$ when the ℓ_2 norm of either arm's six-dimensional clean normalized torque exceeds its frozen threshold; the resulting scalar weight is broadcast to all 12 dimensions. The observed first-frame anchor is removed from both the force errors and their weights. Weighted errors and weights are accumulated within each microbatch before division, and gradient accumulation averages the resulting microbatch-level objectives. With one window per GPU microbatch and $\lambda_c = 3$, high-load steps receive four times the base weight before normalization.

In the 14B InP configuration, the observed image is supplied through image-latent and mask channels rather than by replacing a noisy video latent with a clean value. Consequently, all 13 video latent frames participate in the video-flow loss, whereas the first force token is clamped to its observed value and excluded from the force loss.

Optimization and sampling. The loss uses $\lambda_\tau = 0.1$ and $\lambda_c = 3$, with contact thresholds of 0.349 (left arm) and 0.325 (right arm) in normalized units, fitted on a 400-episode calibration subset of the training split. Training runs for 24k steps at a constant learning rate of 5×10^{-5} with an effective batch of 32 windows on 8 GPUs. For the four-task prediction results in Table 1, ForceWorld uses 50 denoising steps per prediction chunk. Rollouts longer than one window are produced autoregressively: each window is conditioned on the last generated frame, re-encoded by the VAE, and on the model's own last predicted force, and the policy under evaluation reads its next observation from the committed frames.

Torque prediction modes. The main model jointly denoises normalized torque proxies and video latents at a shared diffusion time, using the residual temporal tokenizer, action modulation of force tokens, and contact-phase reweighting. Table 2 compares tokenizer variants and the incremental addition of AMP and PACE.

A.1 Heterogeneity-Aware Mechanism Implementation

The extended action conditioner of AMP emits force-token modulation vectors through a zero-initialized projection so the zero-initialized AMP projection ensures that the AMP branch initially contributes no modulation. TARE (Equation 4) zero-initializes the temporal transformer's output projection so that it degenerates to the framewise embedding at initialization. High-load labels for Equation 5 are computed from ground-truth torque magnitude against frozen per-arm thresholds; λ_c was selected on the validation split ($\lambda_c^* = 3$). In the main-model training recipe, the newly initialized force modules and the Wan backbone are jointly finetuned with a shared learning rate of 5×10^{-5}. No contact classifier or learned router is required for the primary model.

B Evaluation Protocols and Metrics

Torque metrics are reported per joint and macro-averaged across 12 dimensions to prevent high-variance joints from dominating. The high-load stratum threshold and its contact-transition window are fixed on training ground-truth torque, never on model predictions (Section 3.3). They define torque-evaluation strata; they do not assign policy-success labels. The policy outcomes in Section 4.5 are manually labeled.

B.1 Shared Evaluation Details

Hyperparameters, loss weights, and stopping steps are selected on the validation split. The evaluator's training-data sources and separation from policy-evaluation rollouts are described in Appendix D.

Force is scored on denoised predictions, stratified into free motion, contact transition, and contact by thresholds on the ground-truth magnitude (Section 3.3), never on model output. A lightweight arm also registers external torque from a carried payload, from stiction at velocity reversals, and from thermal drift, so the contact stratum is a high-force phase rather than contact in the strict sense. We report free-motion and contact-phase RMSE in proxy units, together with contact-phase

correlation, step-wise contact AUROC, and contact-onset timing error in frames. Video is scored per view against the raw frames with PSNR, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018). Policy evaluation reports Pearson correlation over 12 latest-checkpoint policy–task pairs and MMRV computed within each task and averaged over the four tasks (Appendix D). Evaluation windows are initialized from the real observation and measured torque at the window start; rollout lengths are specified in the corresponding experiment protocols. Force-conditioned variants receive this same anchor, whereas the visual-only counterpart has no force input. Table-specific sampling budgets are given in Appendix B.2. Intervals resample trajectories and then windows, and comparisons between rows are paired per trajectory.

B.2 Prediction-Table Protocols

Every row uses the same windows and seeds. Torque errors are computed separately for each sampling draw and then averaged across draws; onset error is computed over matched events within each run and then averaged over runs with valid matches. The 95% bootstrap intervals resample episodes and then windows. RGB metrics in Table are means. Ctrl-World is retrained on our trajectories as a video-only baseline; TACO is retrained from its released code on the same trajectories.

Contact transitions and event matching. For each evaluated arm, a contact segment is a contiguous run in which the ground-truth six-joint torque magnitude exceeds that arm's frozen threshold. Its first five evaluation frames, or the entire segment if shorter, define the contact-transition phase; the remaining frames define sustained contact. Thus, the transition window starts at contact onset and does not extend into preceding free motion. For combined phase labels, either arm's transition takes precedence over sustained contact, and contact takes precedence over free motion. Event detection retains threshold-exceeding segments lasting at least two frames in both predictions and ground truth. Within each arm, ground-truth events are visited chronologically and greedily matched to the nearest unmatched predicted onset within five frames, with ties resolved in favor of the earlier predicted event. Onset error is the mean absolute onset difference over matched events, measured in evaluation frames. Unmatched ground-truth and predicted events count as false negatives and false positives, respectively, but receive no timing-error penalty. Runs with no matched events have undefined onset error and are excluded from its mean.

Step-wise contact AUROC. The continuous score for each arm and timestep is the ℓ_2 norm of the predicted six-dimensional torque vector, evaluated in the same proxy space as the ground truth. The binary target is whether the corresponding ground-truth norm exceeds the frozen per-arm threshold. Scores and labels are pooled over the evaluated arms within each run; predictions are not thresholded, and the two-frame minimum duration used for event detection is not applied to these step-wise labels. AUROC is computed from ranks with average ranks for tied scores. Runs containing only one target class have undefined AUROC and are excluded from its mean.

A persistence baseline that holds the initially observed torque throughout the prediction window has a contact-phase RMSE of 0.402 on the 77 windows included in its evaluation.

Component ablations. Table 2 uses action conditioning, a single training seed, and 39 Peel cucumber test clips of ten chunks each. Force errors are reported in task-specific proxy units across all twelve joints. The correlation r is scored in the contact phase, AUROC measures step-wise contact detection, and LPIPS is computed on the image canvas. Intervals are 95% bootstraps over clips.

B.3 Matched Visual-Only Control on Peel Cucumber

We compare visual-only and joint video–torque models on Peel cucumber using the same training settings. Table 4 reports right-arm-view video metrics over 39 evaluation windows with sampling seeds 42–43. The two variants have the same reported LPIPS of 0.346; the joint model has slightly lower PSNR and higher SSIM. These results support comparable video quality with and without the torque prediction stream in this matched single-task setting.

Table 4: **Matched visual-only control on Peel cucumber.** Video quality with and without the torque prediction stream under the same training settings.

Torque prediction stream	PSNR↑	SSIM↑	LPIPS↓
Without	**15.47**	0.587	0.346
With	15.36	**0.594**	0.346

C Contact-Constrained Motion Evaluation Details

Episodes and conditions. The reported comparison uses ten real Push-T episodes with a visible approach followed by blocked gripper motion. A rigid object may translate, rotate, or slip; such reference outcomes are recorded separately rather than assumed to be stalls. Both models use the recorded actions and the same rollout horizon.

Visual outcomes. Five annotators watched each complete generated video and visually judged whether the predicted gripper passed into the rigid block. Twenty videos were assessed per model. Apparent penetration was identified in 13 of 20 visual-only videos (65%) and 1 of 20 joint-model videos (5%).

Sampling and statistics. For each of the ten episodes, we generate two complete rollouts per model using the same prediction horizon and denoising budget. We average image metrics over the two rollouts within each episode and compute 95% percentile bootstrap confidence intervals using 10,000 episode-level resamples, with model checkpoints held fixed.

D Policy Evaluation and Human Annotation

The main cross-family evaluation uses one checkpoint per family: ForceVLA, $\pi_{0.5}{+}\tau_{\text{ext}}$, and FAWAM. Each checkpoint is evaluated on Wipe board, Unlock, Peel cucumber, and Screw bulb, with 20 real and 20 imagined rollouts per policy–task pair (240 of each in total). Imagined runs use predicted torque feedback.

Recorded success counts. In task order Wipe board, Unlock, Peel cucumber, and Screw bulb, ForceVLA has real success counts $(15, 8, 5, 5)$ and imagined counts $(17, 11, 4, 6)$; $\pi_{0.5}{+}\tau_{\text{ext}}$ has $(16, 10, 4, 7)$ and $(17, 12, 7, 9)$; FAWAM has $(13, 5, 3, 2)$ and $(13, 7, 2, 5)$. Every denominator is 20.

Additional checkpoints of $\pi_{0.5}{+}\tau_{\text{ext}}$. Figure 7(b) compares Checkpoint 1, Checkpoint 2, and Checkpoint 3 of $\pi_{0.5}{+}\tau_{\text{ext}}$ across the same four tasks. Checkpoint 3 uses the results reported above. In the same task order, Checkpoint 1 has real success counts $(14, 5, 3, 3)$ and imagined counts $(15, 10, 6, 5)$; Checkpoint 2 has $(17, 7, 3, 3)$ and $(16, 12, 4, 3)$. Each denominator is 20. The unweighted Pearson correlation across these 12 checkpoint–task pairs is $r = 0.9410960$.

Initial conditions. For each policy–task pair, we conduct 20 real-world trials and 20 corresponding imagined trials. Scenes are manually reset before each real-world trial. We record the initial RGB observations, joint state, and torque feedback from that trial and use them to initialize its imagined counterpart. This provides paired initial observations for real and imagined evaluation.

Closed-loop execution. In both environments, each policy query returns 50 actions, $\mathbf{a}_{0:49}$, where $\mathbf{a}_0$ is aligned with the current observation. The world model uses the current robot state for its anchored first frame and $\mathbf{a}_{1:48}$ for the following 48 generated frames. The final action $\mathbf{a}_{49}$ is discarded when the policy replans. Before being supplied to the world model, the actions are converted to end-effector poses using forward kinematics. Actions are executed at 10 Hz, with periodic replanning based on updated observations, while the policy retains its joint-space state representation. The policy then generates a new action chunk from these updated inputs. Both real and imagined trials run to a preset duration rather than terminating immediately upon success.

Table 5: Real and imagined policy performance within each task. Entries report real / imagined success rates (%), each measured over 20 trials. MMRV is computed separately for each task on the $[0, 1]$ success-rate scale.

Task	ForceVLA	$\pi_{0.5}+\tau_{\text{ext}}$	FAWAM	Task MMRV ↓
Wipe board	75 / 85	80 / 85	65 / 65	0.0167
Unlock	40 / 55	50 / 60	25 / 35	0.0000
Peel cucumber	25 / 20	20 / 35	15 / 10	0.0333
Screw bulb	25 / 30	35 / 45	10 / 25	0.0000

Task-success criteria. We assign a binary outcome to each rollout using the same task-specific criteria in real and imagined evaluation. For *Wipe board*, success requires removing all marks to be erased from the board. For *Unlock*, success requires picking up the key, aligning it with the keyhole, and opening the door. For *Peel cucumber*, success requires completing the initial peeling pass at one end of the cucumber, rotating the cucumber, and completing a second peeling pass. For *Screw bulb*, success requires picking up and aligning the bulb, tightening it into the socket, and lighting it.

Annotation and aggregation. Five human annotators review each complete rollout video with the policy identity hidden and assign its binary outcome. For each policy–task pair and evaluation environment, the success rate is the number of fully successful trials divided by 20.

Evaluator–policy firewall. The world-model checkpoint is held fixed throughout policy evaluation. The evaluator is trained on teleoperated demonstrations, including unsuccessful attempts, and excludes rollouts from the evaluated policies. The failed trajectories retained in the mixture (Section 4.1) are teleoperated collection failures from the demonstration corpus.

Statistics. For each of the $n = 12$ policy–task pairs, let x_i and y_i be real and imagined success counts divided by 20. We compute unweighted Pearson correlation across these pairs ($r = 0.9556156$). For each task, MMRV averages each policy's maximum real-success-rate gap over inconsistent pairwise orderings between real and imagined evaluation. Using strict pairwise comparisons, the task-averaged MMRV is 0.0125. Specifically, with task index t and policy indices i, j,

$$V_t(i,j) = |x_{t,i} - x_{t,j}|\, \mathbf{1}[(y_{t,i} > y_{t,j}) \neq (x_{t,i} > x_{t,j})]\,,$$
$$\text{MMRV} = \frac{1}{4}\sum_{t=1}^{4}\frac{1}{3}\sum_{i=1}^{3}\max_{1\le j\le 3} V_t(i,j). \tag{7}$$

The plotted ordinary least-squares fit is $y = 0.988832x + 0.075161$. Pearson correlation summarizes agreement across policy–task pairs, while MMRV measures within-task ranking consistency among the three evaluated policies.

D.1 Within-Task Policy Comparison

To assess policy discrimination beyond differences in task difficulty, we compare real and imagined success rates within each task (Table 5).The task-averaged MMRV is 0.0125.

D.2 Sensitivity to Test-Time Torque-Feedback Replacement

We use test-time torque-feedback replacement to examine how the fixed force-aware policies respond when their torque observations are suppressed. The policy checkpoints are held fixed without retraining; feed-zero replaces the torque feedback supplied during imagined execution with zeros. This intervention changes the torque input distribution relative to the contact-dependent feedback used in policy training and real execution.

Table 6 reports the feed-zero success counts for all 12 policy–task pairs, with 20 imagined trials per pair. Using the real success counts reported above as the reference, the unweighted Pearson correlation is $r = 0.8879965$, and the ordinary least-squares fit is $y = 1.071066x - 0.052538$. All

Table 6: Imagined policy success with test-time zero torque feedback. Entries report success rates (%) over 20 trials. Policy checkpoints are identical to those used for the main evaluation and are not retrained.

Task	ForceVLA	$\pi_{0.5}+\tau_{\text{ext}}$	FAWAM
Wipe board	70	85	55
Unlock	60	60	40
Peel cucumber	10	5	10
Screw bulb	10	15	15

12 pairs contribute separately to these statistics. The lower correlation under feed-zero describes sensitivity to this feedback intervention.

D.3 Manual Outcome Annotation

Task success in both real and imagined rollouts is determined by human inspection. Figure 7 uses these manual labels directly. Predicted torque is returned to the policy as feedback, which is distinct from the procedure used to label the resulting rollout.

E Limitations

Our force representation uses current-derived joint-torque residuals to reflect changes in joint loading during interaction with the environment. Its scale and joint-space structure depend on the robot platform. Extending this representation to calibrated Cartesian wrenches and incorporating tactile signals would support finer-grained analysis of grasp forces and incipient slip.

Our current validation focuses on manipulation tasks on a single bimanual platform. Generalization across platforms and new interaction scenarios remains an open direction. Broadening the coverage of robot embodiments, objects, and contact conditions would enable further study of transfer in joint visual and force modeling, as well as its use for ongoing policy evaluation and checkpoint selection during training.

F Qualitative Rollouts Across the Four Tasks

Figure 8 presents qualitative examples of successful and failed imagined rollouts across four tasks.

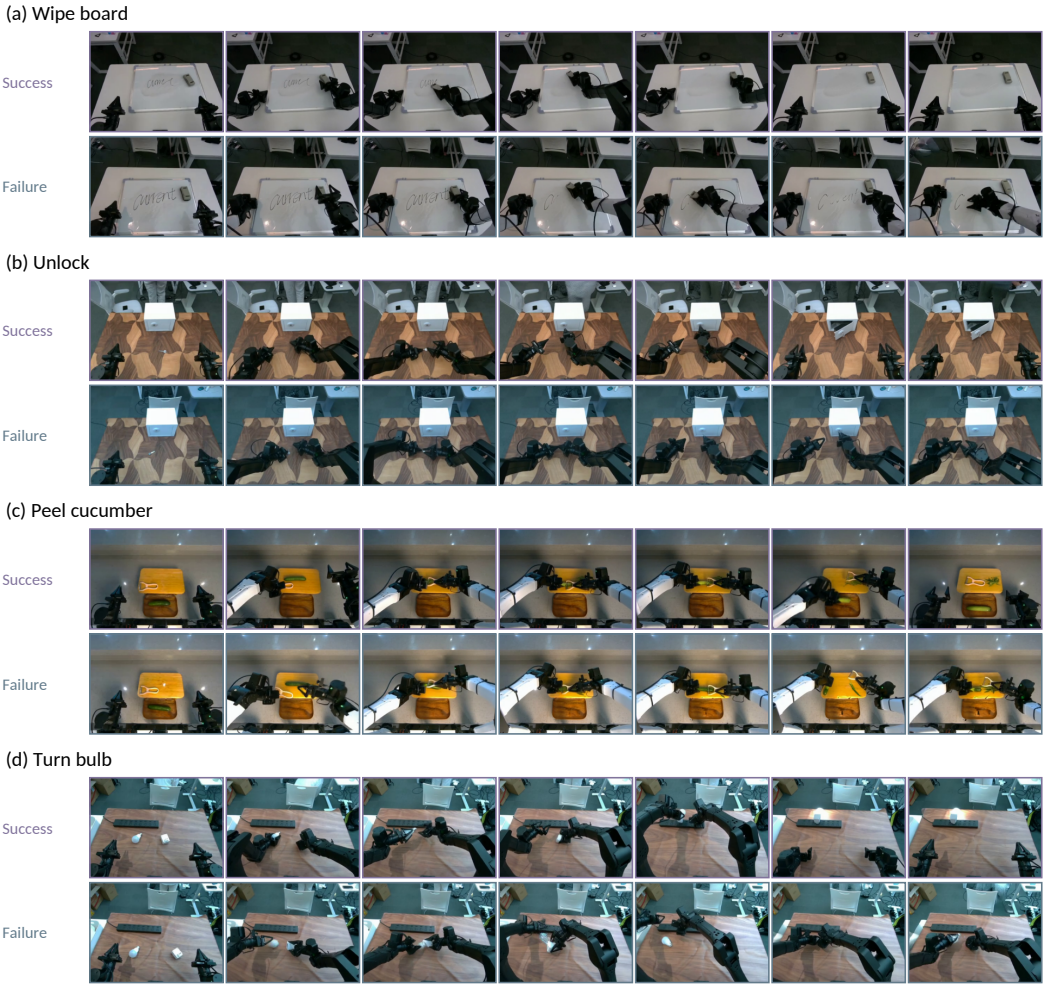


Figure 8: **Successful and failed rollouts across four tasks.** Each row shows seven main-camera frames from a single trial, ordered chronologically from initialization to the illustrated outcome and sampled at nonuniform intervals. **(a) Wipe board:** the successful trial clears the writing, while the failed trial leaves marks on the board. **(b) Unlock:** the successful trial opens the cabinet door, while the failed trial leaves it closed. **(c) Peel cucumber:** the successful trial removes peel and returns the cucumber to the tray in the final frame, while the failed trial leaves the task unfinished with the cucumber on the working surface. **(d) Screw bulb:** the successful trial ends with the bulb installed and illuminated, while the failed trial continues manipulating the bulb and socket without completing installation.