$\mathcal{N}_0$-TWAM: Scaling Tactile-Native World Action Model for Contact-Rich Manipulation

NeoteAI Team & Fudan TEAI Team

We present $\mathcal{N}_0$-TWAM, a tactile-native world-action model for contact-rich manipulation that predicts both future vision and contact. To our knowledge, it is the first tactile world-action model trained at large scale, and it demonstrates strong capability on contact-rich tasks. We pre-train $\mathcal{N}_0$-TWAM at large scale with visuo-tactile joint training over tactile-rich demonstrations, spanning six embodiments and 450 tasks. We use NeoForce, a unified force-based tactile representation to form a physically grounded contact signal as condition for action generation. To boost long-horizon and multi-stage manipulation, we introduce tactile contact events for task staging and advance through them during execution. For real-time efficiency, we adopt an asymmetric Mixture-of-Transformers architecture on $\mathcal{N}_0$-TWAM, which pairs a full-width expert for versatile videos and slim architecture for downstream action and tactile experts. Evaluations on both real and simulated benchmarks justify the powerful capabilities of $\mathcal{N}_0$-TWAM in various contact-rich tasks, as well as demonstrate the benefit of data scaling on precise tactile and action prediction. In summary, $\mathcal{N}_0$-TWAM endows a world-action model with predictive capabilities to foresee vision, tactile and action, building a solid foundation for fine-grained manipulation on open contact-rich tasks. The codebase and model checkpoints of $\mathcal{N}_0$-TWAM will be made publicly available to foster further research and development in tactile-enabled robotic manipulation.

Date: July 25, 2026
Code: https://github.com/neoteai/N0-TWAM
Website: https://research.neoteai.com/n0-twam/

NeoteAI
Institute of Trustworthy Embodied AI, Fudan University

1 Introduction

Learning a general manipulation policy is hard in contact-rich settings. Seating a screw, peeling one cup off a stack, or closing a gripper on a soft object are decided less by the global layout of the scene than by what is happening at the fingertips, a distinction that a camera can rarely resolve but that a force or tactile sensor reports directly. A policy that is to act well in these regimes therefore needs two things that pure visuomotor regression does not naturally provide: access to touch, and the ability to *anticipate* how the immediate future of an interaction will unfold rather than reacting only to the present frame.

Two recent lines of work each supply one of these ingredients. Vision-language-action (VLA) models attach an action head to a large pretrained vision-language model and map observations to actions, inheriting broad semantic priors but compressing a rich visual present into a low-dimensional action with no explicit model of what happens next [8]. Video world models, on the other hand, learn to roll the future of an observation forward in time, and a growing body of *world-action* models couples such a video predictor with action so that control is grounded in a predicted future [45, 82]. The latter framing is attractive for contact-rich control (if the model can predict the next moment, it can decide against it), yet the predicted future of these models is almost always purely visual.

Touch is beginning to enter this picture, but so far along three partial paths. Tactile policies *consume* touch as an input channel and never predict it [36]. A second family forecasts touch with a separate (often frozen) tactile predictor bolted onto the policy, so prediction occurs happens outside the network that acts [80, 87]. And concurrent touch-aware world-action models admit touch into the model but keep it at arm's length, gating or masking tactile tokens to protect the visual stream [52, 70, 94]. On every path, touch is not part of what the model predicts. What is missing is a model that predicts touch together with vision, under the same objective and at the same causal step, and a policy that reads its actions from that jointly predicted future.

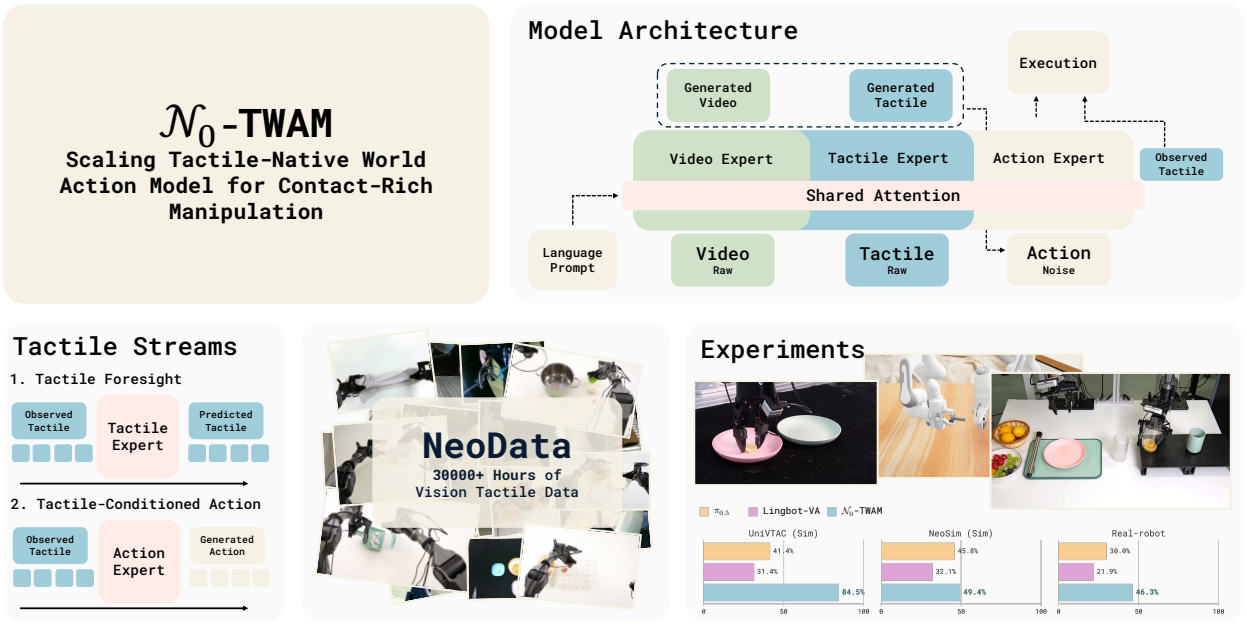

Figure 1 Overview. $\mathcal{N}_0$-TWAM is pre-trained on large-scale self-collected real-robot manipulation data with synchronized per-finger tactile streams. An MoT with three per-modality experts (video, tactile, and action) predicts the future vision and tactile, then predicts the action from them, so touch is both *predicted* (a predicted future) and *observed* (an observed present). Across contact-rich benchmarks in simulation and on real robots, $\mathcal{N}_0$-TWAM outperforms prior world-action and vision-language-action baselines.

We close this gap with $\mathcal{N}_0$-TWAM (Tactile World-Action Model), a world-model policy and, to our knowledge, the first tactile world-action model trained at large scale. $\mathcal{N}_0$-TWAM is built on a video-diffusion transformer backbone and keeps the world-action recipe of jointly modeling future observations and action, but it predicts *tactile as part of the predicted future* rather than treating it as a side input. The backbone is reorganized as a Mixture-of-Transformers (MoT) [50]: three per-modality experts (video, action and tactile) each keeps their own weights and capacity and interact only through a single shared self-attention at every layer. Where concurrent touch-aware models protect their visual stream by restricting touch in *attention*, $\mathcal{N}_0$-TWAM isolates capacity in *weights*, so vision and touch stay fully attentive to one another. A frame-id causal schedule turns this shared attention into a "predict-then-act" cascade inside one forward pass: within a chunk the video and tactile experts co-generate the future scene and the future contact, and the action expert is conditioned on those just-predicted quantities. That future contact is the predicted tactile; as it acts, the action expert also reads the observed tactile, cross-attended in from the current reading. The policy therefore acts on touch it has already predicted, grounded in the touch it feels now.

Treating three modalities as full experts would, however, triple the cost of the largest component, and the action and tactile streams have no large-scale pretraining to justify that cost. $\mathcal{N}_0$-TWAM instead uses an *asymmetric* design: the video expert keeps its full, original width, while the action and tactile experts are slim and trained from scratch, benefiting from the visual prior only indirectly through the shared attention. This halves the parameter count relative to an all-full-width design, and the same asymmetry keeps inference cheap: once a chunk's video and tactile are predicted, their attention keys and values are cached, so each action-denoising step re-runs only the lightweight action expert rather than the full-width video expert.

Across simulated and real contact-rich benchmarks, $\mathcal{N}_0$-TWAM is the strongest method overall: it reaches 84.5% on UniVTAC and 49.4% on NeoSim in simulation, and averages 46.3% across our eight-task real-robot suite, against 30.0% for the strongest vision-language-action baseline and 21.9% and 14.4% for the vision-only world-action baselines. Ablations attribute the gains to both tactile roles (removing either the predicted foresight target or the observed conditioning degrades success), and under distribution shift the policy degrades more gracefully than vision-only world-action baselines. We summarize our contributions as follows.

- **A native tactile world-action model (§2.1).** We present $\mathcal{N}_0$-TWAM, a world-action model with touch as

a *native* modality of its MoT backbone: a dedicated tactile expert predicts future touch jointly with future vision under one shared self-attention, rather than as a side input. We pre-train it at scale on tens of thousands hours of real-robot data (much of it carrying synchronized tactile), spanning six embodiments and 450 tasks, collected with our own system.

- **A dual-pathway predictive tactile design (§2.2).** We model touch inside the world-action model along two pathways: a *predicted* tactile stream the model generates as a foresight target, and an *observed* tactile stream that conditions action generation. In post-training, the observed pathway reads touch through *NeoForce* [56], effectively extracting force representation on manipulation tasks.
- **A tactile-aware sub-task system for long-horizon manipulation (§2.3).** We use tactile contact events, such as a grasp or a release, to segment long-horizon demonstrations into sub-task-conditioned clips for training and to schedule execution stages at inference: the predicted tactile triggers the advance to the next sub-task, and the observed contact event confirms it, so both tactile roles drive the planner.
- **Strong empirical results (§4).** $\mathcal{N}_0$-TWAM achieves state-of-the-art results on contact-rich benchmarks in simulation and on real robots. Our analyses trace these gains to touch itself: both tactile pathways contribute, performance grows with more pre-training data, and it stays robust under distribution shift.

2 Model

$\mathcal{N}_0$-TWAM is a world-action model: rather than regress actions from observations, it learns one generative model that rolls the future the robot will see and feel forward in time and reads the action out of that predicted future. Three choices set it apart from a video world-action model. (i) *Touch is modeled jointly with vision*: the future tactile signal is generated under the same flow-matching objective as the future video, at the same causal step, so the model predicts touch directly rather than deriving it from pixels. (ii) *Capacity is isolated in weights, not attention*: each modality gets its own expert, and the experts share a single self-attention, so vision and touch keep private weights yet stay fully attentive to one another; concurrent touch-aware models instead gate or mask tactile tokens to protect the visual stream. (iii) *Touch plays two roles*: it is both predicted ahead of time, as a foresight target, and observed now, as an observation the action reads. We set up the architecture (§2.1), develop the two tactile roles (§2.2), and close with tactile-aware execution (§2.3).

2.1 Model architecture

Problem setup. Conditioned on a language instruction and the observation history, $\mathcal{N}_0$-TWAM predicts three coupled streams of the robot's near future: what it will *see*, what it will *feel*, and the *action* it will take. Given the observation history and the instruction c, $\mathcal{N}_0$-TWAM predicts each chunk's future video x_k^v, tactile x_k^t, and action x_k^a, and models their joint distribution autoregressively over chunks:

$$p_\theta\big(x^v_{1:K},\, x^t_{1:K},\, x^a_{1:K} \,\big|\, c\big) \;=\; \prod_{k=1}^{K} \underbrace{p_\theta\big(x^v_k,\, x^t_k \,\big|\, X_{<k},\, c\big)}_{\text{predict}} \cdot \underbrace{p_\theta\big(x^a_k \,\big|\, x^v_k,\, x^t_k,\, X_{<k},\, c\big)}_{\text{act}}, \tag{1}$$

where the observation history $X_{<k} = (x^v_{<k}, x^t_{<k}, x^a_{<k})$ collects the past video, tactile, and actions. $\mathcal{N}_0$-TWAM treats vision and touch as equally important and predicts the two jointly. The noisy action token is denoised conditioned on the just-predicted x_k^v and x_k^t of its own chunk, so the policy acts on an anticipated future.

Conditional flow-matching objective. Flow matching [51] regresses the velocity of the straight path from a clean target x to Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$. We apply it at *latent-frame* granularity. Let $x^m_{k,j}$ denote temporal group j of modality $m \in \{v, a, t\}$ in chunk k: a VAE latent frame for video and tactile, and the action steps aligned with each latent frame for action, so $J_v = J_t = J_a = 2$. For each group we independently sample $\sigma^m_{k,j} \sim p(\sigma)$ and $\epsilon^m_{k,j} \sim \mathcal{N}(0, I)$:

$$\hat{x}^m_{k,j} \;=\; (1 - \sigma^m_{k,j})\, x^m_{k,j} \;+\; \sigma^m_{k,j}\, \epsilon^m_{k,j}, \qquad u^m_{k,j} \;=\; \epsilon^m_{k,j} - x^m_{k,j}. \tag{2}$$

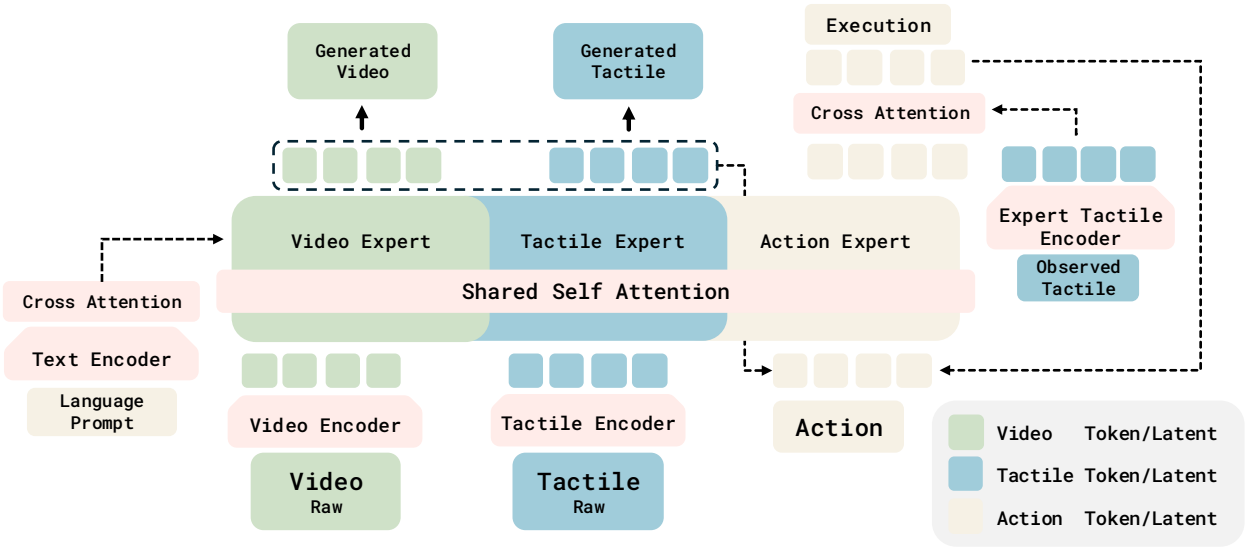

Figure 2 Overview of $\mathcal{N}_0$-TWAM. Multi-view RGB and tactile streams are encoded into latent tokens and processed by three per-modality experts (*video*, *tactile*, *action*) that interact only through a single shared self-attention at every layer, ordered by the frame-id causal cascade: the video and tactile experts co-generate the predicted future scene and future contact, and the action expert denoises the action chunk conditioned on the just-predicted pair. The language instruction enters the video and action experts through cross-attention, but not the tactile expert. A lightweight observed pathway (right) encodes the *observed* current tactile and cross-attends into the action tokens before the action head (§2.2); the denoised actions are decoded and executed.

Within a group, the scalar $\sigma^m_{k,j}$ is shared by all its tokens (the spatial tokens of a video or tactile latent frame, or the aligned action steps) while the entries of $\epsilon^m_{k,j}$ remain independent. Writing $\hat{X} = \{\hat{x}^m_{k,j}\}$ and $\Sigma = \{\sigma^m_{k,j}\}$, the loss is

$$\mathcal{L} \;=\; \sum_{m\in\{v,a,t\}} \frac{\lambda_m}{J_m} \sum_{j=1}^{J_m} \mathbb{E}\Big[\Big\| \sqrt{w^m_{k,j}} \odot \big(f^m_{\theta,k,j}(\hat{X}, \Sigma, c; \mathcal{M}) - u^m_{k,j}\big)\Big\|_2^2\Big], \tag{3}$$

where $w^m_{k,j} \geq 0$ combines SNR weighting with the action-validity mask that drops the absent arm on single-arm embodiments.

Mixture-of-Transformers backbone. Following the MoT design [50], we split the backbone into three *experts*, one per modality, that share *only* a single self-attention at each layer (Fig. 2). At layer ℓ, expert m normalizes its own hidden state, applies its own timestep modulation, and projects into a common attention width d:

$$Q^m, K^m, V^m \;=\; \mathrm{Proj}^m_\ell\big(\mathrm{AdaLN}^m_\ell(h^m)\big), \qquad m \in \{v, a, t\}. \tag{4}$$

The per-expert queries/keys/values are concatenated in the fixed order $[\,v\,|\,a\,|\,t\,]$ and run through one masked self-attention,

$$[\,O^v\,|\,O^a\,|\,O^t\,] \;=\; \mathrm{Attn}\big([\,Q^v|Q^a|Q^t\,],\, [\,K^v|K^a|K^t\,],\, [\,V^v|V^a|V^t\,];\, \mathcal{M}\big), \tag{5}$$

after which each expert applies its own gated output projection and feed-forward network.

Parameter-efficient experts. A naive three-expert backbone would triple the cost of its largest component. We instead keep only the shared attention at full width d and let each expert run a narrower private residual/FFN width d_m: thin boundary projections leave the concatenated attention unchanged while the FFN, norms, and residual stream run at d_m. The full-width *video* expert is warm-started from a pretrained non-MoT, shared-backbone world-action model [45] reorganized here into experts, so the backbone inherits a strong visual and dynamics prior. The slim *action* and *tactile* experts run at $d_a{=}d_t{=}1024$ against the video expert's $d_v{=}3072$; not matching the shared width, they are trained from scratch and reach the visual prior

only through the shared attention. Because the video expert alone is about 70% of the backbone, slimming the two new modalities rather than widening them roughly halves the trainable model (Table 1).

Table 1 Trainable MoT backbone configuration. The asymmetric design keeps a full-width video expert (warm-started from a pretrained video model) alongside slim action and tactile experts (trained from scratch), roughly halving the parameters of an all-full-width variant. Shared self-attention runs at d=3072 (24×128 heads) across all layers; "Hidden Size" is each expert's private residual/FFN width d_m. The frozen Wan VAE and umT5 text encoder sit outside the backbone and are excluded; all-full-width figures are estimates.

		Video		Action		Tactile		
Model	# Layers	Hidden Size	Params	Hidden Size	Params	Hidden Size	Params	Total Params
$\mathcal{N}_0$-TWAM (asym.)	30	3072	5.00 B	1024	1.13 B	1024	1.03 B	7.16 B
All-full-width	30	3072	5.00 B	3072	≈5.0 B	3072	≈5.0 B	≈15 B

Diffusion-forcing cascade. The shared attention is governed by a causal mask $\mathcal{M}$ that realizes the factorization of Eq. (1) inside one forward pass. Two indices order every token: a latent-frame position and a noise state. Within a chunk, its two latent frames take consecutive positions; vision and touch tokens from the same frame share one position, and the action follows the predicted frames on which it conditions. In state, each token inherits the noise level of its latent-frame group and is *noisy* while it is being generated and *clean* once it has become history. At training time these clean history tokens are the ground-truth past: the model is *teacher-forced* on the true history and only denoises the current chunk, following the diffusion-forcing regime [16] in which independent temporal-group noise levels generalize teacher forcing to continuous tokens. As a robustness measure, the clean video and tactile history is lightly re-noised with probability 0.5, so the model learns to tolerate a committed history that at test time is imperfect rather than ground truth; the action history stays exactly clean. The mask then follows three rules: (i) a clean token attends to clean tokens at or before its position; (ii) a noisy token attends to the clean history strictly before its position; and (iii) a noisy token attends to noisy tokens at its *own* position, i.e. co-generation.

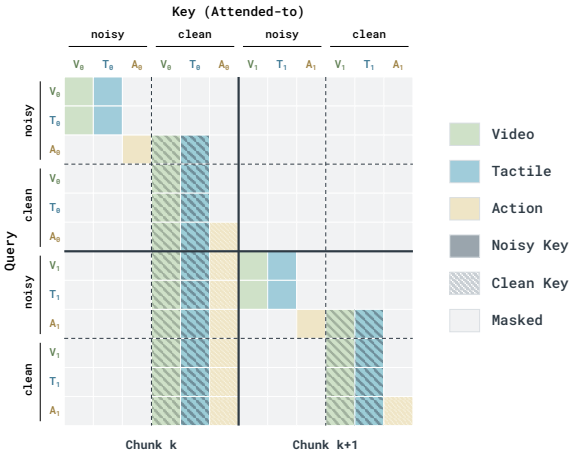

Figure 3 Joint self-attention mask over two chunks, with each V/T block schematically collapsing the two latent frames. Each token appears as a key twice, noisy (solid) and clean (hatched); rows are queries, columns keys, colored by the key's modality. Within a chunk the noisy video and tactile co-generate with each other, and the noisy action additionally attends to their clean copies; each new chunk attends to the whole clean history, and no token ever attends a future chunk.

Figure 3 visualizes this mask for two chunks. The predict-then-act cascade of Eq. (1) is thus produced entirely by the mask, with no change to the attention operator itself. Because every temporal group draws its noise level independently, training covers all combinations of clean and noisy groups; the two patterns used at inference (a clean committed history with a noisy current chunk, and clean predicted frames with a still-noisy action) are simply two of these combinations.

Action representation. Each arm contributes a 10-dimensional per-frame action (a 3-D end-effector position, a 6-D orientation [95], and a 1-D gripper command), so a bimanual action is 20-dimensional; single-arm embodiments fill one half and the rest is masked out of the loss. Following $\pi_{0.5}$ [7], we regress *delta* end-effector targets rather than absolute poses, anchored to the chunk-start pose, forming the delta by plain element-wise subtraction: $\Delta p = p_t - p_a$ and $\Delta r = r_t - r_a$, a coordinate-wise difference rather than a relative rotation $R_a^\top R_t$ on SO(3). We prefer subtraction because a relative rotation is a nonzero constant when the pose is nearly stationary, so a small prediction error accumulates into a persistent orientation offset, whereas subtraction

makes the rest state the zero vector $\Delta r{=}0$, the natural target of a regressor. This keeps the targets small and centered wherever the arm is; the gripper command stays absolute, and this 20-D space is the unified action interface used for pre-training across all embodiments.

2.2 Modeling touch

The tactile sensors we use are *vision-based*: a tactile sensor is essentially a small camera imaging a soft gel that deforms on contact, so a tactile observation is itself an RGB video stream. $\mathcal{N}_0$-TWAM therefore treats scene video and touch as equally important modalities: the same VAE encodes them and the same generative objective predicts them.

Contact states such as a secure grasp, incipient slip, or excessive force are decisive for the next action yet often invisible to the external scene cameras. $\mathcal{N}_0$-TWAM therefore gives touch two roles: a future the model *predicts* (§2.2.1) and a present the policy *reads* as it acts (§2.2.2).

2.2.1 Predicting future touch

Each tactile stream is VAE-encoded [65] and patchified with the same $(1, 2, 2)$ patch as the video latent, so touch enters the backbone as ordinary latent tokens; a learned *sensor-id* embedding (up to four sensors) lets one expert serve single-arm and bimanual data unchanged. Sharing the latent space with video is what makes the shared attention meaningful, since vision and touch tokens become directly comparable. Touch nonetheless keeps *private weights*: tactile statistics are sparse and event-driven, and concurrent touch-aware world models report that injecting such tokens into a visual dynamics model degrades video and action prediction [70]. Isolating touch at the *weight* level, rather than restricting it in *attention*, lets vision and touch stay mutually attentive while keeping separate capacity. The tactile expert also does not cross-attend to text.

The tactile expert treats the *predicted* tactile latent as a generation target alongside video: its two latent frames are independently noised and optimized by Eq. (3); this makes the model predict future contact a chunk ahead rather than only reading the current tactile signal. We predict future touch in *residual* form. Let x_i^t be the tactile latent at future step i and x_0^t the initial (step-0) frame; the tactile expert predicts the residual

$$\Delta_i^t \;=\; x_i^t - x_0^t, \qquad \text{so that} \qquad x_i^t \;=\; x_0^t + \Delta_i^t, \tag{6}$$

rather than the absolute frame, since touch is near-constant away from contact and its informative content is the change Δ_i^t at contact onset and release. Each tactile token takes the temporal position of the video frame it accompanies, so it obeys the same causal and windowing rules with no tactile-specific bypass. Two consequences follow. First, the predicted scene and predicted contact are denoised *together* under rule (iii) of §2.1, each conditioning the other, so the predicted video and predicted contact stay consistent. Second, a noisy tactile token reads only committed history, never a same-step clean copy of itself, so no separate leak-guard rule is needed once touch is aligned to the same frames as the video.

2.2.2 Conditioning on observed touch

Touch reaches the action expert in *two* representations, each matched to its role (Fig. 4). The *predicted* pathway is anticipatory: the predicted future contact flows to action through the cascade, and because it must be *generated* it lives in the VAE latent space shared with video. The *observed* pathway is reactive, and the action head need only *understand* the current reading, not generate it, so we read it in *force space*. A frozen estimator maps the raw tactile image to a dense three-axis surface force map $[f_x, f_y, f_z]$ with a contact mask, a physically grounded, sensor-agnostic reading of contact; a NeoForce tactile encoder [56], fine-tuned end-to-end with the action loss, turns this map into representation tokens. Those tokens are cross-attended into the action stream just before the action head, under a chunk-causal mask:

$$h^a \;\leftarrow\; h^a \;+\; \mathrm{CrossAttn}\big(h^a,\; E_{\mathrm{obs}}(x^t_{\mathrm{now}})\big), \qquad W_{\mathrm{out}} = 0 \text{ at initialization}, \tag{7}$$

so the pathway starts as an exact no-op and grafts onto a pretrained model without perturbing it. In short, $\mathcal{N}_0$-TWAM *predicts* touch in latent space and *observes* it in force space. The observed-tactile encoder E_{obs} is representation-agnostic, and we instantiate it per domain. On real robots the tactile sensor matches the one NeoForce was pretrained on, so E_{obs} is the NeoForce force-space encoder. In simulation the tactile

comes from a different sensor that NeoForce never saw, so we do not read it in force space; instead E_{obs} is a lightweight branch trained from scratch on the current-frame tactile latent: a linear patch embedding with learned sensor-id and frame/height/width position embeddings whose tokens enter the same zero-initialized cross-attention port, with no force estimator and no pretrained representation.

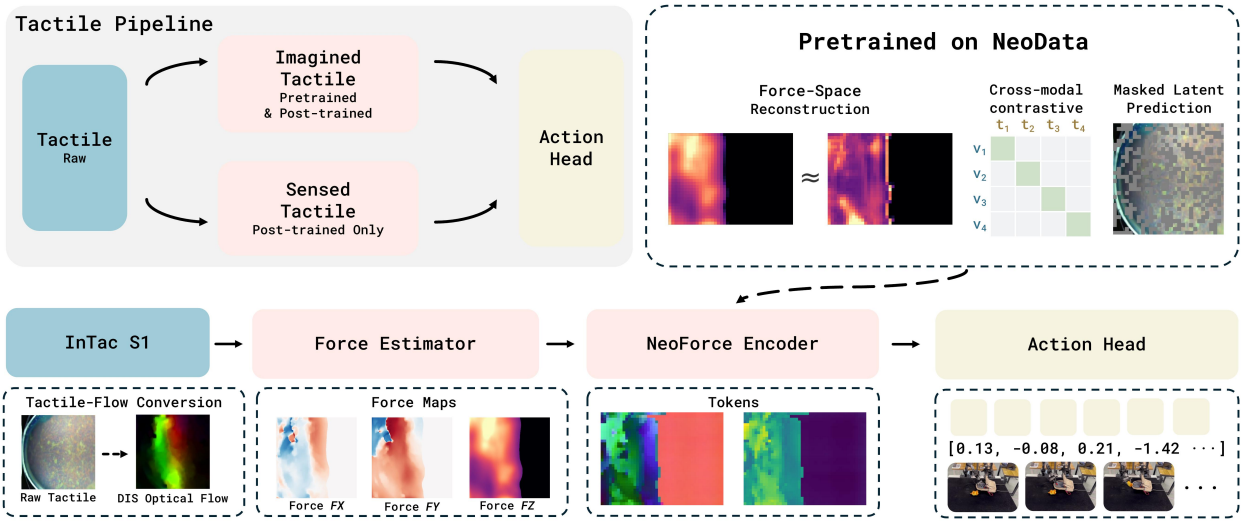

Figure 4 The two tactile pathways. Touch enters $\mathcal{N}_0$-TWAM in two ways. As a *predicted* signal, it is generated in the VAE latent space shared with video and denoised by the tactile expert as a foresight target. As an *observed* signal, it is read in force space: a frozen estimator g_θ maps the raw tactile image to a three-axis force map $[f_x, f_y, f_z] \rightarrow$ a NeoForce [56] tactile encoder (warm-started, fine-tuned) produces representation tokens $\rightarrow$ cross-attended into the action stream through the zero-initialized port of Eq. (7). An optional reconstruction term anchors the fine-tuned representation to the force field.

Staged training. We separate the two roles across the two training stages (§4.1). Pre-training uses *only* the predicted stream, with the observed pathway off: its job is to learn a generative world model at scale, and the predicted tactile latent is a generation target that fits that objective, forcing the model to predict future contact jointly with the future scene and so learn contact dynamics across the large-scale corpus. The observed stream does not fit that objective. It is a conditioning input rather than a generation target, it depends on the pretrained NeoForce encoder and on paired current-tactile readings, and, most importantly, giving the policy the true current contact during pre-training would let it rely on that signal and under-train the foresight it is meant to complement. Post-training then switches on the observed pathway of Eq. (7), adding it on top of a model that has already learned to predict future contact; the ablations of §4.5 separate the two contributions.

The pretrained force-space representation (NeoForce). The encoder of the observed pathway is *NeoForce*, the tactile representation model of $\mathcal{N}_0$-Foundation [56], summarized here only as far as $\mathcal{N}_0$-TWAM needs it.

NeoForce takes a synchronized chunk of RGB observations and tactile force maps and returns a tactile representation in *force space*. Its tactile input is not a raw gel image but a dense three-axis surface force map $\hat{F} = g_\theta(T) \in \mathbb{R}^{H \times W \times 3}$ produced by a learned estimator g_θ. The in-plane components $[f_x, f_y]$ carry the shear that drives sliding and grasping, the normal component f_z carries the pressure of pressing, and a binary contact mask marks where the surface is touched, giving six channels for a parallel gripper. Because these units are physical rather than tied to the appearance of a particular gel, one encoder can serve different sensors [84]. The visual and tactile streams are patchified independently and fused by a shared ViT backbone initialized from DINOv2 weights, so vision and touch are read in a common space and the representation captures how contact evolves across the chunk. A reconstruction head decodes the force map and contact mask back out, which keeps the representation tied to a physical quantity.

$\mathcal{N}_0$-TWAM imports both stages and treats them differently. The estimator g_θ is a calibrated sensor-to-physics converter and stays *frozen*, so gradients stop at the force map. The NeoForce encoder warm-starts

E_{obs} in Eq. (7) and is *fine-tuned* with the action loss, which lets the policy specialize contact priors learned at scale to the behaviors it is learning. Its reconstruction head comes along to supply the optional force-space anchor noted above.

2.3 Tactile-aware execution

At test time $\mathcal{N}_0$-TWAM runs autoregressively over chunks: within each chunk it follows the cascade of Fig. 3, denoising the future video and tactile latents, then the action conditioned on those just-predicted quantities, then decoding and executing before sliding on. The autoregressive rollout, its rolling key/value cache, and the asynchronous prediction pipeline are standard machinery for video world-action models. What is specific to $\mathcal{N}_0$-TWAM is that inference is *tactile-aware*: touch enters the closed loop on three time scales at once, a reflex at sensor rate (the observed pathway of Eq. (7)), a prediction at chunk rate (the tactile foresight denoised alongside video), and a task clock at event rate (the tactile-punctuated scheduler of §2.3.3).

2.3.1 Streaming and deployment

The rolling cache commits in the frame-id causal order of training, with an attention window aligned to the training window. Following observation-grounded streaming of the video stream, we extend the same grounding to *touch*: as real frames *and* real tactile readings arrive, they replace the corresponding predicted entries, so the rolling context stays anchored to what the robot actually saw and felt on one timeline. The observed tactile pathway of Eq. (7) sits outside the denoising loop, so its conditioning can be refreshed at sensor rate between chunk-level re-predictions. The first chunk has no committed history, so its first frame is clamped to the observed initial frame, and the paired action is the zero delta of the chunk-start anchor; alternatively, the first frame can be left free and denoised from noise like any other frame.

2.3.2 Real-time inference

Three properties of $\mathcal{N}_0$-TWAM enable its inherited rolling generation, caching mechanism, and asynchronous pipeline to operate in real time even at the 7B-parameter scale: (1) The cascade imposes a one-way dependency within each action chunk: neither the video queries nor the tactile queries attend to the actions in the current chunk. Consequently, once the prediction phase has completed denoising the future video and tactile signals, their per-layer keys and values remain unchanged throughout the subsequent action loop. The asymmetric MoT architecture makes this strategy computationally effective: we cache these keys and values and *only* re-run the lightweight action expert at each action denoising step. (2) The predicted video and tactile latents share one denoising-step schedule, advancing under the same scalar timestep at every step of the prediction loop, while the action uses an *independent* schedule in its own loop. The action branch can therefore adopt a shorter denoising schedule without incurring the per-step computational cost of the video expert. (3) Across successive chunks, a rolling history cache reuses previously committed chunks, while the asynchronous pipeline predicts the next chunk as the robot executes the current one. As long as the computation for the next chunk fits within the execution window of the current chunk, the inference latency can be hidden behind robot execution. As an optional deployment-time configuration, the model backbone can also be served at reduced precision. The linear layers in each expert operate in FP8, or NVFP4 on Blackwell-class hardware, while numerically sensitive operations (including the attention QKV projections and softmax, normalization layers, and output heads) remain in bf16. This strategy reduces the memory footprint and matrix-multiplication cost of the dominant video expert while having a negligible effect on the action output. The additional computational cost introduced by the tactile modality is negligible. The predicted contact signal is generated in the same forward pass that already performs scene prediction, while the observed branch in Eq. (7) consists only of a lightweight cross-attention module that is refreshed at sensor rate outside the denoising loop.

2.3.3 Long-horizon execution via tactile punctuation

In long-horizon tasks, a single episode-level instruction does not tell the model *which stage of the task* the robot is currently in. Mid-task observations are often stage-ambiguous (a table halfway through "make a cup of tea" looks much like the table before pouring), the attention window does not span the episode, and a constant prompt carries no information about the current stage, so the text pathway goes unused *within* a

task and the policy stalls, repeats a step, or skips ahead. The missing structure already lives in the tactile stream: sub-tasks are delimited by *contact events* (a grasp begins at contact onset and ends at release; a pour begins when the vessel is loaded and ends when it is set down), between which touch is steady and at which it changes sharply. The tactile stream is naturally *punctuated*, and in $\mathcal{N}_0$-TWAM touch is already a stream the model observes and predicts, so we let the same signal structure the task on both sides of training (Fig. 5).

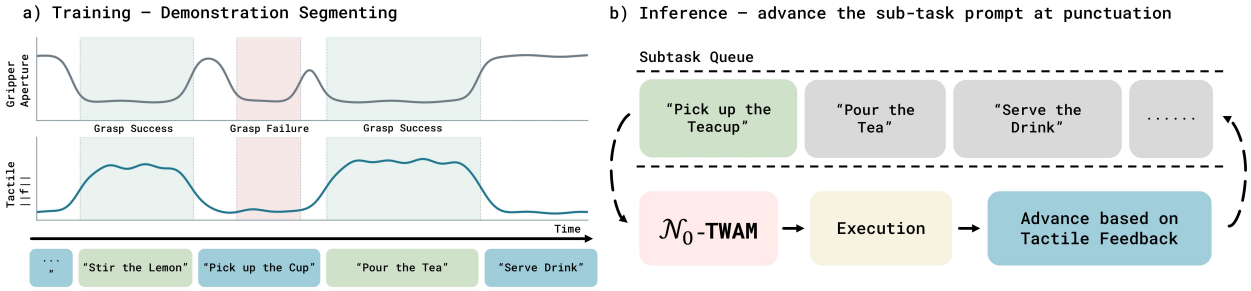

Figure 5 Tactile-punctuated long-horizon execution. *(a) Training:* a long-horizon demonstration, drawn as gripper aperture and tactile magnitude $\|f\|$ over time, is segmented at contact events into sub-task clips. The tactile trace also reports whether each grasp *succeeds*: a gripper can close without loading any force, so a failed grasp shows an aperture dip but no tactile rise, and the one signal that delimits sub-tasks also confirms them. Each clip is labeled with a short instruction ("Pick up the Cup", "Pour the Tea", ...) and post-trained through the existing text pathway. *(b) Inference:* a lightweight scheduler holds the sub-task queue; $\mathcal{N}_0$-TWAM runs on the current sub-task prompt and executes it, and when tactile feedback signals the sub-task is complete the scheduler advances to the next prompt, looping through the queue.

Segmenting demonstrations. We segment long-horizon demonstrations using tactile "punctuation," primarily detecting contact-onset and release events from changes in tactile signals, with gripper aperture used as a fallback when necessary. The full demonstration is thereby divided into multiple sub-task clips. This approach extends keyframe-based imitation learning [60] to *contact events*. Because a grasp produces a rise in tactile signals only after the gripper has actually made contact with an object, the same tactile reading can also distinguish a successful grasp from a gripper that closes on nothing (Fig. 5a). Thus, a single signal is sufficient both to segment the demonstration and to verify task success. We use Gemini 3.5 Flash [19] to generate a short sub-task instruction for each clip, with human spot checks, and during post-training each clip is conditioned on its own corresponding sub-task instruction.

Advancing sub-tasks at inference. Touch is not only observed and predicted by the model, but also connected to the inference-time planner. At inference, a lightweight scheduler maintains a queue of sub-tasks and advances to the next prompt when the current sub-task completes (Fig. 5b). Completion combines the model's two tactile roles: the predicted future tactile signal triggers the advance one step early, and the corresponding release or contact-onset event in the observed tactile stream then confirms it before the switch is committed. The sub-task sequence can be fixed in advance for repeatable tasks or generated by a vision–language planner in open-ended settings [1, 48]; in either case, the tactile-event detector decides when to advance from the actual execution, and each advance re-seeds the streaming context with the next prompt, matching the per-clip conditioning used in training.

3 Data

$\mathcal{N}_0$-TWAM is trained on *NeoData* [56], and this section covers only what the model consumes and how it is produced. We first summarize the slice of the dataset we use (§3.1), then the offline pipeline that turns raw episodes into the latent tokens the model operates on (§3.2).

3.1 Dataset

NeoData is a large multi-robot manipulation corpus of over 30,000 hours, spanning six embodiments and 450 contact-rich tasks, in which a large fraction of the episodes carry synchronized per-finger tactile alongside multi-view RGB. Tactile coverage at this scale is what separates it from vision-only pre-training corpora, and is what lets $\mathcal{N}_0$-TWAM learn to predict contact rather than only appearance. We refer to the NeoData report [56] for the collection system, sensors, curation, and full statistics; the real-robot post-training episodes use the InTac S1 visuotactile sensors described in §4.2.

3.2 Data processing

Latent encoding. To train efficiently at the scale of 10^9 frames, we encode both the camera views and the tactile streams with a *single* frozen causal video VAE [65], treating each tactile stream as a small video so that touch shares the latent space of vision, and we encode the language instruction with a frozen umT5 text encoder. All of these inputs are converted to latents once during preparation and cached to disk, so the encoders never run in the training forward pass.

Chunking. Each episode is segmented into fixed windows of 33 latent frames, the unit over which the chunked cascade of §2.1 operates. A window spans 387 raw frames at 30 fps, subsampled to 129 frames at 10 fps and compressed to 33 latent frames by the VAE's 4× temporal downsampling. Consecutive windows advance by a stride of 24 latent frames, leaving a ∼27% overlap that keeps segment junctions covered, and tail windows are right-aligned to the episode end. A one-time validation pass records which windows are complete across all required modalities, and only complete windows are used for training.

Unified action space. To train one policy across heterogeneous robots, every embodiment's action is canonicalized into a single 20-dimensional end-effector space, 10 per arm. Every embodiment is recorded with end-effector pose; the position and 6D-rotation targets are formed as the chunk-anchored deltas of §2.1 while the gripper is kept absolute, and actions are normalized *per robot* rather than globally, with quantile clipping on the heavy-tailed gripper channel.

Contact-event stage segmentation. For the long-horizon pipeline of §2.3.3 we also mark *stage* boundaries within a demonstration, with the *tactile* stream as the primary signal: most transitions in contact-rich tasks (first contact, loss of contact, slip, an insertion seating) show up as a sharp change in the tactile reading. Where the tactile cue is weak or ambiguous, we fall back to the gripper aperture, which reliably marks grasps and releases, and a final human check corrects the remaining boundaries. This keeps the sub-task labeling and inference-time scheduling of §2.3.3 keyed on meaningful contact events.

4 Training and Experiments

4.1 Implementation Details

Pre-training. We pre-train the world-action model on tens of thousands hours of manipulation, from the multi-robot corpus of §3: real-robot demonstrations across six embodiments (ARX5, UR, Flexiv, Franka, PiPER and hand-collected UMI) data. Each clip provides multi-view RGB pre-encoded with the Wan2.2 VAE at 4× temporal and 16× spatial compression, a synchronized predicted tactile stream, and the 20-D bimanual delta end-effector action. The training objective is the multi-modal flow-matching loss of Eq. (3): video, touch, and action are trained with equal 1:1:1 modality weights; every latent-frame group is noised independently, with one noise level shared across a frame's spatial tokens for video and tactile, and across the action steps aligned with that frame for action. At this stage tactile enters *only* as the predicted foresight target (§2.2), while the observed tactile pathway is disabled, so pre-training concentrates capacity on predicting future vision and contact at scale. We follow a two-stage curriculum: the model is first pre-trained on the real-robot data to convergence, then continued from that checkpoint with UMI data mixed in (about 60% of the batch). The video expert is a ∼5B video-diffusion transformer of the Wan2.2-TI2V-5B family [65], warm-started from

LingBot-VA [45], while the action and tactile experts are trained from scratch. We optimize with AdamW at a learning rate of 10^{-4} and weight decay 0.1 for 30,000 steps, about 2.2 epochs, at an effective batch of 512. Training runs on 128 NVIDIA H800 GPUs under FSDP2 with bf16 mixed precision, and the optimizer updates only the MoT backbone since the tokenizer and text encoder are frozen. We hold out 3% of each task's data as a validation split and score a random subset of it at each validation pass.

Tactile encoder (NeoForce). The observed tactile pathway (§2.2.2) uses a ViT-B backbone initialized from DINOv2 [59] and shared across the patchified visual and tactile streams, and each input is a chunk of 4 synchronized RGB frames and tactile force maps. We first pre-train NeoForce on 20,000 real-robot and Neo TacUMI demonstrations spanning 30 tasks [56], for 100,000 steps on 8 NVIDIA A100 GPUs at a learning rate of 2×10^{-5}. $\mathcal{N}_0$-TWAM then loads that checkpoint and fine-tunes the encoder with the action loss during post-training.

Post-training. For each downstream setting, we adapt the pre-trained $\mathcal{N}_0$-TWAM checkpoint to task-specific demonstrations while preserving the multi-modal flow-matching cascade. Each demonstration is segmented into synchronized windows of multi-view RGB, language, action, and tactile, with RGB and language pre-encoded as Wan-VAE latents and text embeddings so that optimization focuses on the transformer and tactile-conditioning modules. The action is chunked at a horizon of 24 steps, 12 per latent frame. The observed-tactile encoder differs by domain: on real robots it is the NeoForce force-space encoder above, while in simulation, where the signal is not produced by a real sensor, it is a lightweight encoder trained from scratch.

Observed tactile readings condition action prediction (§2.2): the current-frame tactile latent is embedded by the zero-initialized observed pathway and cross-attended into the action expert under a chunk-causal mask, so these clean condition tokens are visible to action queries, never enter the video stream, and are not themselves a denoising target; the pretrained force-space head of §2.2 plugs into the same port. In parallel, predicted tactile latents remain a denoising target under the cascade, so the action expert draws on both predicted future contact and observed current contact.

Condition dropout. During training we randomly drop the language and tactile conditions, each with probability 0.1, by two different mechanisms. Dropping language follows the standard classifier-free recipe, replacing the text embedding with a learned null embedding. Dropping tactile instead uses *absence* rather than zeroing: on a dropped sample no tactile tensor is passed at all, so the tactile tokens do not appear in the attention sequence, the tactile modules receive no gradient, and that step trains the pure video–action model. This absence design has three consequences. First, it unifies deliberate dropout with genuinely tactile-less data: episodes from embodiments without tactile sensors train through exactly the same interface. Second, the model learns the marginal, vision-only policy alongside the tactile-conditional one, so if a sensor fails or is absent at deployment the policy degrades to its vision-only self instead of reacting to an out-of-distribution zero input. Third, it exposes a classifier-free-guidance knob at inference: the gap between the tactile-conditional and unconditional action predictions measures, and can amplify, how much the policy listens to touch. In all reported evaluations we leave this knob at w=1 (no tactile CFG amplification); the vision-only fallback it enables is probed in §4.4.

Optimization. Post-training data use the LeRobot dataset format, and the real-robot configuration uses three RGB cameras and up to four tactile streams. The objective sums the video-latent, predicted-tactile, and masked-action denoising losses at equal 1:1:1 weights. We fine-tune all transformer parameters, including the observed tactile encoder, with AdamW at a learning rate of 10^{-4} and weight decay 0.1, bf16 mixed precision, and gradient clipping at 2.0; task-specific runs use one sample per GPU, four-step gradient accumulation, and 1000 optimizer steps unless noted.

4.2 Evaluation Protocol

Benchmarks and metric. We evaluate on three suites: UniVTAC [15], a public tactile-manipulation simulation benchmark (eight tasks); NeoSim [56], our in-house simulation suite (twelve tasks, four single-arm and eight

Table 2 Success rate (%) on the UniVTAC simulation benchmark (eight tasks). Methods are grouped into vision-language-action (VLA) policies and world-action models (WAM); the best per column is in **bold**.

Method	Insert HDMI	Insert Hole	Insert Tube	Grasp Classify	Lift Can	Lift Bottle	Pull-out Key	Put Bottle in Shelf	Avg.
Vision-language-action (VLA)									
$\pi_{0.5}$ [7]	8	25	74	49	6	**100**	35	34	41.4
StarVLA-α [81]	24	52	69	68	65	32	51	**88**	56.1
InternVLA-A1 [11]	12	93	95	86	90	58	66	37	67.1
Xiaomi-Robotics-0 [12]	**69**	96	**98**	45	13	21	**80**	12	54.3
World-action models (WAM)									
GigaWorld-Policy [79]	0	12	9	20	0	38	32	21	16.5
LingBot-VA [45]	38	42	96	17	0	0	58	0	31.4
FastWAM [86]	19	66	**98**	72	0	21	73	35	48.0
$\mathcal{N}_0$-TWAM (Ours)	68	**99**	**98**	**94**	**93**	58	79	87	**84.5**

dual-arm); and a real-robot suite of eight real-world contact-rich manipulation tasks, four on the dual-arm PiPER and four on the single-arm Flexiv, six of which are drawn from NeoReal [56]. We report task success rate (%) and its task average (a macro average, weighting every task equally regardless of trial count); the best per column is **bold**. Each simulation task is evaluated over 100 trials and each real-robot task over 20 trials, every trial starting from a randomized initial configuration; the reported success rate is the fraction of successful trials, with success given by the benchmark-defined completion check in simulation and by a fixed per-task criterion judged by the operator on the real robots.

Closed-loop action decoding. All methods are evaluated closed-loop under the asynchronous receding-horizon protocol of §2.3. At each control cycle the policy consumes the current multi-view RGB together with the rolling history cache and the current observed-tactile latent, runs the cascade once to predict the future scene and contact, and denoises a single action chunk—16 steps of the unified 20-D bimanual delta end-effector action—with the slim action expert on its own short denoising schedule. Chunks are executed asynchronously, the robot running the current chunk while the next is predicted, and the observed-tactile condition is refreshed at sensor rate between chunk-level re-predictions; a trial terminates on task success or when it reaches a per-task step budget of 1.5× the length of that task's training demonstrations.

Baselines. We compare $\mathcal{N}_0$-TWAM against two kinds of baselines: world-action models and recent strong vision-language-action (VLA) models. The world-action baselines are LingBot-VA, the unified video world-action backbone our model builds on [45]; FastWAM, a MoT world-action model [86]; and GigaWorld-Policy, an efficient action-centered world-action model [79], all vision-only without tactile. The VLA baselines are $\pi_{0.5}$ [7], StarVLA-α [81], InternVLA-A1 [11], and Xiaomi-Robotics-0 [12]. Coverage varies by suite and each figure and table lists the methods it includes: $\pi_{0.5}$, the strongest VLA, is run on all three suites.

Real-robot tasks and sensor. All real-robot experiments use the **InTac S1** visuotactile sensor (Shanghai Xinzhi Embodied Intelligence Technology Co., Ltd. / NeoteAI), mounted on two embodiments: a dual-arm PiPER and a single-arm Flexiv. Our post-training suite has eight contact-rich tasks (Fig. 8), each with tactile-critical decision points (secure grasp, incipient slip, first contact, jamming, support transfer, final seating) that external cameras often cannot resolve. The dual-arm PiPER runs Cup Stacking, Board Wiping, Making Lemon Tea, and Bag Packing; the single-arm Flexiv runs Fruit Collection, Bottle Standing, Socket Plugging, and Hanoi Tower. Making Lemon Tea and Hanoi Tower are long-horizon tasks (§2.3.3).

4.3 Main Results

Following the evaluation protocol of §4.2 (benchmarks, metric, and baselines), we report per-benchmark results below, with the best per column in **bold**.

Simulation on UniVTAC. On the public UniVTAC tactile-manipulation benchmark (Table 2) we compare against vision-language-action (VLA) policies and world-action models (WAM).

$\mathcal{N}_0$-TWAM reaches the highest average success (84.5%), about 17 points above the strongest baseline of any kind (InternVLA-A1, 67.1%). Notably, the vision-only world-action baselines trail even the VLA policies here (LingBot-VA 31.4, FastWAM 48.0, GigaWorld-Policy 16.5): predicting the future scene is not enough for tasks defined by contact. Making tactile native to the world-action model is what closes that gap, lifting $\mathcal{N}_0$-TWAM past the VLA policies and the vision-only world-action baselines alike.

Simulation on NeoSim. Table 3 reports per-task results on NeoSim, our in-house simulation suite.

Table 3 Per-task success rate (%) on the NeoSim simulation suite (four single-arm and eight dual-arm tasks). Methods are grouped into vision-language-action (VLA) policies and world-action models (WAM); the best per row is in **bold**. FastWAM and the ACT baseline were not run on NeoSim.

	Vision-language-action (VLA)				World-action models (WAM)		
Task	$\pi_{0.5}$	StarVLA-α	InternVLA-A1	Xiaomi-Robotics-0	GigaWorld-Policy	LingBot-VA	$\mathcal{N}_0$-TWAM
Single-arm							
Insert USB	74	72	36	62	34	31	**100**
Grasp Chip	86	**93**	12	49	71	77	92
Unplug & Plug Charger	**23**	0	0	5	3	0	0
Pour Ball	**92**	32	47	52	0	37	63
Dual-arm							
Bowl Unstack	3	17	0	0	0	18	**72**
Cup Stack	**22**	0	0	0	12	20	12
Insert Screw	26	8	0	0	0	18	**29**
Place Gears	**18**	0	0	0	0	0	0
Cup Handover	13	**25**	0	0	0	0	14
Plate Stack	96	0	0	0	0	93	**98**
Bowl Stack	93	0	8	42	0	67	**97**
Cup Unstack	3	31	0	**71**	9	24	16
Average	45.8	23.2	8.6	23.4	10.8	32.1	**49.4**

On NeoSim, $\mathcal{N}_0$-TWAM (49.4 average) is the best method overall, ahead of the strongest baseline $\pi_{0.5}$ (45.8), and leads by a clear margin on the contact-heavy stacking and insertion tasks (Insert USB, Bowl Unstack, Plate Stack, Bowl Stack). The tactile advantage is smaller here than on real robots: the simulated tactile is not the sensor $\mathcal{N}_0$-TWAM was pretrained on, so its observed pathway falls back to the lightweight sim encoder rather than the NeoForce force-space representation (§2.2.2). The real-robot results below are where native touch matters most.

Real-world results. Figure 7 reports task-level success on our eight-task real-robot suite, six of them drawn from NeoReal. We compare against $\pi_{0.5}$, the strongest VLA policy in our simulation experiments, and the world-action baselines LingBot-VA and FastWAM.

Overall ordering. Across the eight real-robot tasks, $\mathcal{N}_0$-TWAM is best overall, with a macro average of 46.3% against 30.0% for $\pi_{0.5}$, 21.9% for LingBot-VA, and 14.4% for FastWAM. $\pi_{0.5}$ is the strongest baseline, and LingBot-VA, which retains the full video world-action prior, comes next, while the efficiency-oriented FastWAM trails both. The margin is not uniform. It is largest exactly where success hinges on a hidden

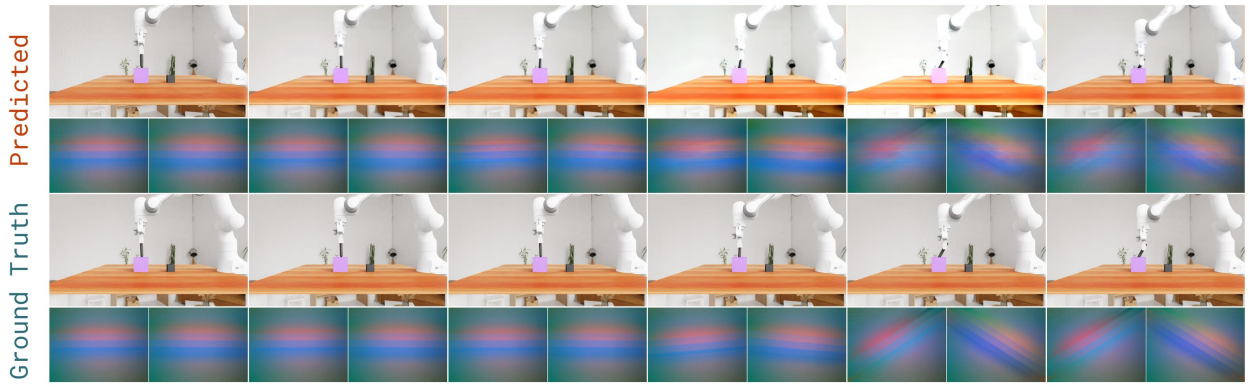

Figure 6 Qualitative simulation rollout (Insert Hole): the model's *predicted* future (top) against the *ground-truth* rollout (bottom). In each block the multi-view RGB scene sits above its tactile stream; the predicted scene and contact track the real execution across the insertion.

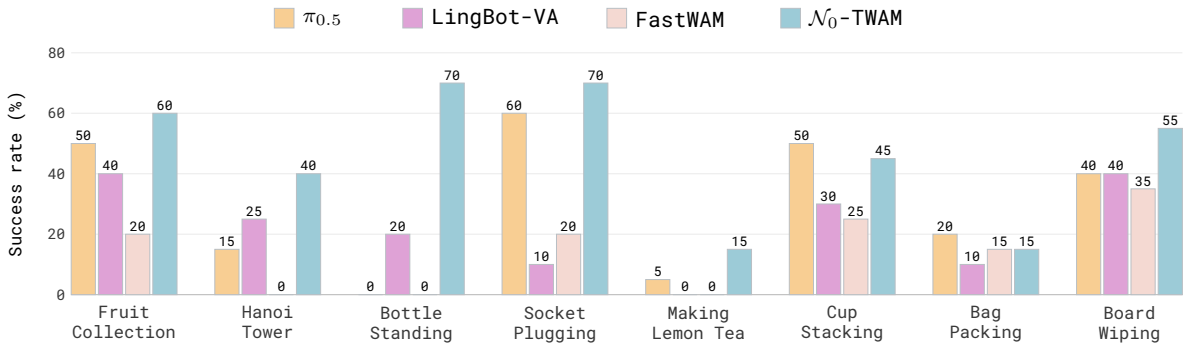

Figure 7 Per-task success rate (%) on the real-robot suite, eight contact-rich tasks, four methods each (bar value on top; $\mathcal{N}_0$-TWAM in blue). The methods are the vision-language-action policy $\pi_{0.5}$ [7] and the world-action models LingBot-VA [45] and FastWAM [86]. Macro average (equal weight per task): $\mathcal{N}_0$-TWAM 46.3, $\pi_{0.5}$ 30.0, LingBot-VA 21.9, FastWAM 14.4. Six of the eight tasks are drawn from the NeoReal benchmark [56].

contact state a camera cannot resolve: standing a bottle upright by observing its weight shift (Bottle Standing, 70 vs. 20 for the best baseline), seating a plug (Socket Plugging, 70 vs. 60), a secure fine grasp (Fruit Collection, 60 vs. 50), and wiping under sustained contact load (Board Wiping, 55 vs. 40). On the two tasks whose outcome is set by visually guided placement rather than contact, $\pi_{0.5}$ matches or edges $\mathcal{N}_0$-TWAM (Cup Stacking, 45 vs. 50; Bag Packing, 15 vs. 20), consistent with the gain coming from touch rather than from task length alone. Each real-robot number is a success rate over 20 trials, so a per-task rate carries a binomial standard error of up to $\sim$11%; we therefore read individual per-task gaps as indicative and rely on the consistent direction across tasks and the task-averaged margin.

Why $\mathcal{N}_0$-TWAM leads. $\mathcal{N}_0$-TWAM retains the strong video–action prior, but adds touch to the control loop in two complementary forms. The predicted tactile expert co-denoises future contact with future video, and the causal cascade lets the action expert read this committed tactile future before producing the next action (§2.2). This anticipatory path helps the policy avoid actions whose visual outcome appears plausible but whose predicted contact is inconsistent with a stable grasp or insertion. In parallel, the observed pathway injects the current force-space tactile representation directly into the action expert (§2.2.2), providing a short-latency correction signal for slip, shear, asymmetric contact, and excessive pressure. The two paths therefore address different failure modes: tactile foresight reduces errors before contact is committed, whereas observed touch supports recovery after contact begins.

The implementation preserves the pretrained policy while adding these signals. Touch uses private expert weights and exchanges information with vision and action only through shared attention, preventing sparse,

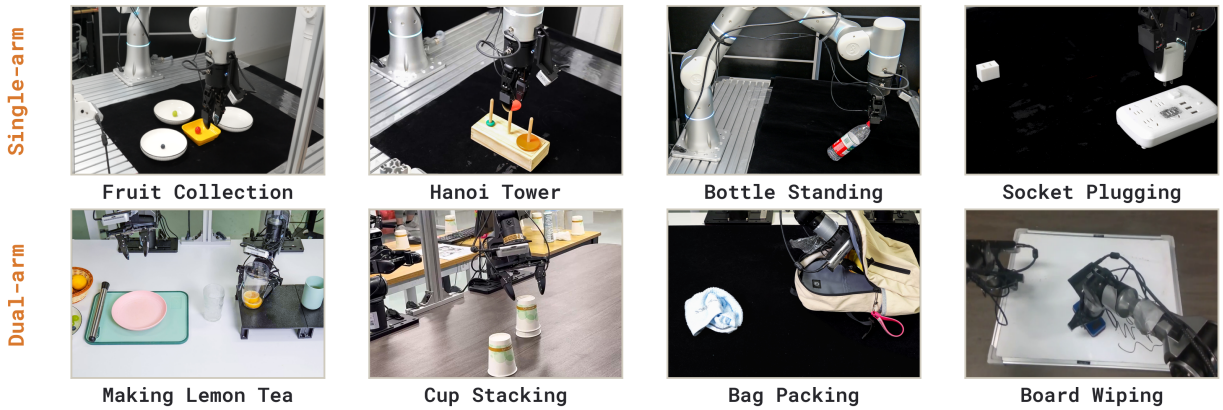

Figure 8 The real-robot task suite. Eight contact-rich tasks on two embodiments: four single-arm tasks on a Flexiv (top row) and four dual-arm tasks on a PiPER (bottom row). Six of the eight are drawn from NeoReal [56].

event-driven tactile statistics from overwriting the visual representation. The staged recipe first learns predicted tactile foresight at scale and activates the observed tactile reflex only during task adaptation, preventing the reactive signal from becoming a shortcut that replaces predictive contact modeling. The observed cross-attention output is zero-initialized, so post-training starts from the pretrained behavior rather than an abruptly perturbed policy. Finally, tactile-condition dropout trains the same action expert both with and without tactile tokens, which limits over-reliance on touch during free-space motion and makes the policy robust to temporarily missing contact observations. These design choices let post-training add contact sensitivity without sacrificing the scene understanding inherited from the base world-action model.

Task-level gains. The same mechanism explains where the advantage is largest. For Fruit Collection and Bottle Standing, tactile feedback balances secure acquisition against object deformation, slip, and a shifting center of mass. Board Wiping exposes sustained load and support transfer against the surface. Socket Plugging and Hanoi Tower place the strongest demand on both pathways: observed force direction supplies a correction signal after rim or peg contact, while tactile foresight helps distinguish a recoverable alignment from an impending jam or unstable seating. In the long-horizon tasks (Making Lemon Tea and Hanoi Tower), these contact events also provide reliable boundaries for the tactile-punctuated execution mechanism of §2.3.3, reducing the chance that one failed transition corrupts the remaining action sequence. The two tasks where touch adds least, Cup Stacking and Bag Packing, are also the two whose outcome is set by visually guided placement, which is why the vision-strong $\pi_{0.5}$ stays on par or ahead there.

Baseline analysis. $\pi_{0.5}$ is the strongest baseline. As a large vision-language-action policy it brings broad semantic priors and dependable visually guided placement, which keeps it close to $\mathcal{N}_0$-TWAM on placement-dominated tasks and ahead on Cup Stacking and Bag Packing; but it regresses actions directly from the current frame, with no model of the next moment and no touch, so it can neither anticipate nor feel a contact event. Among the world-action models, LingBot-VA ranks above FastWAM: it shares the strong visual foundation and retains explicit future-video generation for reliable scene-level planning, whereas FastWAM's efficiency-oriented inference collapses the video branch into a single-pass encoder rather than preserving the iterative future rollout [86]. None of the three baselines has a tactile pathway, so visually similar contact states ("aligned but not inserted" versus "fully seated") stay hard to tell apart. The ordering thus points to two cumulative benefits: explicit iterative prediction separates LingBot-VA from FastWAM, and adding both predicted and observed touch inside that prediction–action loop is what carries $\mathcal{N}_0$-TWAM past every baseline, including the vision-strong $\pi_{0.5}$. The component ablations in Figure 9 test this attribution separately from the aggregate baseline comparison.

4.4 Generalization

We probe generalization on our real-robot suite along three axes, each isolated on the task where it matters most (Table 4). *Unseen object*: on Bottle Standing and Bag Packing we keep the task and workspace fixed but replace the manipulated objects with instances never seen in training, bottles of new shape, size, and material, and new items to pack, so success requires transferring the contact skill to novel geometry and physical properties rather than memorizing specific instances. *Unseen position*: on Cup Stacking we use the same cups but initialize them at placements and offsets outside the training distribution, testing whether the stacking behavior generalizes spatially rather than replaying a fixed trajectory. *Visual perturbation*: on Fruit Collection we leave the task and physics unchanged but perturb the visual scene at test time (lighting and background changes), stressing the visual pathway. Because tactile reports contact directly, $\mathcal{N}_0$-TWAM should lean on touch and degrade more gracefully than vision-only world-action models as the visual input shifts.

Table 4 Generalization success rate (%) on our real-robot suite; the best per column is in **bold**.

Method	Unseen Object	Unseen Position	Visual Perturbation	Avg.
$\pi_{0.5}$ [7]	**80**	**45**	25	50.0
LingBot-VA [45]	75	35	30	46.7
$\mathcal{N}_0$-TWAM (Ours)	65	**45**	**45**	**51.7**

The three axes separate cleanly. On *unseen objects* $\pi_{0.5}$ leads (80): its large vision–language pretraining carries broad object and semantic priors, so it adapts to new instances best, whereas the world-action models lean on learned dynamics and transfer less to novel appearance. On *unseen positions* $\mathcal{N}_0$-TWAM matches $\pi_{0.5}$ (both 45) and both clear LingBot-VA. The decisive axis is *visual perturbation*: as lighting and background shift, $\pi_{0.5}$ and LingBot-VA fall to 25 and 30, but $\mathcal{N}_0$-TWAM holds at 45, because tactile reports contact directly and gives the policy a vision-independent signal to fall back on. $\mathcal{N}_0$-TWAM is therefore not the strongest on every axis but the most *robust*: it is highest on average (51.7) and degrades most gracefully exactly where vision becomes unreliable, the regime touch is meant to cover.

4.5 Ablation Study

We ablate the main design choices of $\mathcal{N}_0$-TWAM: the scale of the pre-training data, the *predicted* tactile foresight target, and the *observed* tactile conditioning of the action expert. The 20% *pre-training data* variant matches the full model in architecture, task demonstrations, and post-training schedule, but warm-starts from a checkpoint pre-trained on only a 20% subset of the corpus rather than the full run, isolating how much the downstream gains scale with pre-training data. The two tactile variants remove one pathway each through a single switch, starting from the full model that keeps both. The *observed* pathway is a side branch that cross-attends the current tactile into the action stream, so disabling it simply skips instantiating that branch. The *predicted* pathway is instead coupled into the backbone as both self-attention segments and the tactile expert's generation target, so rather than delete it we apply an *information ablation*: we zero the predicted tactile latent before it is noised, leaving the sequence layout, attention mask, and diffusion loss token-for-token identical to the full model and changing only the information the pathway carries. All variants warm-start from the same checkpoint and share data, hyperparameters, and step count. Each variant is evaluated on UniVTAC and NeoSim (the 20% pre-training variant on UniVTAC only); Figure 9 reports the task-averaged success on each and their mean, with the per-task breakdown in Appendix A. On UniVTAC, pre-training scale is the largest single factor: warm-starting from a 20% checkpoint lowers success from 84.5 to 65.4. Both tactile pathways are load-bearing on top of that: removing the predicted foresight target lowers UniVTAC success to 71.8 and removing the observed conditioning to 70.5, and the NeoSim ablation shows the same ordering more sharply (from 49.4 down to 41.1 and 29.6), so touch contributes in both its predicted and its observed form rather than through a single mechanism.

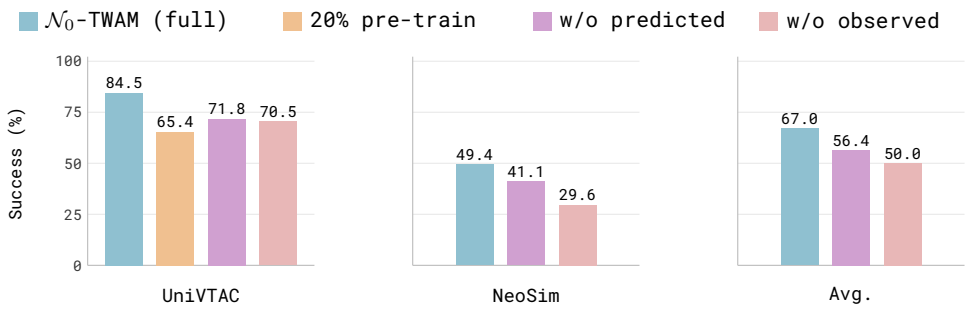

Figure 9 Ablation of $\mathcal{N}_0$-TWAM on UniVTAC (eight tasks), NeoSim (twelve tasks), and their mean. Removing either tactile pathway lowers success, most sharply on NeoSim; the 20% pre-training variant (UniVTAC only) shows pre-training scale is the largest single factor. Per-task breakdowns are in Appendix A.

4.6 Analysis

Delta vs. absolute end-effector actions. We default to the chunk-anchored *delta* end-effector parameterization of §2.1, which is translation-invariant and keeps the targets small and centered, so it transfers cleanly across embodiments and workspace placements. It is not always the best choice at deployment. On alignment-sensitive tasks the absolute end-effector pose can outperform the delta target (Table 5): a delta is measured from the moving chunk-start anchor, so small per-chunk errors accumulate and the target is never tied to a fixed location, whereas an absolute pose grounds each target directly in the world frame. The two parameterizations therefore trade generalization against precision: delta for translation-invariant transfer, absolute for reaching a specific pose. We keep delta for the unified multi-embodiment policy and adopt absolute on the task-specific settings where sub-centimeter placement dominates success.

Table 5 Delta vs. absolute end-effector parameterization on four alignment-sensitive tasks (success rate %). The absolute pose, which we adopt for these tasks, avoids the drift the delta target accumulates.

Parameterization	Lift Can	Lift Bottle	Put Bottle in Shelf	Grasp Chip	Avg.
Delta	24	52	86	38	50.0
Absolute	**93**	**58**	**87**	**92**	**82.5**

Realism of the simulated tactile image. Both tactile images in this study come from the *same* contact simulation and differ only in how the deformed gel is imaged. The UniVTAC numbers above use the simulator's *clean* output, a calibration-based GelSight render (Taxim [61]) that maps the simulated contact-height field to a smooth, deterministic RGB image through a per-pixel illumination model, so contact appears as a clean color-and-intensity blob. A real vision-based sensor looks nothing like this; it returns a silvery, speckled, softly shaded gel image with sensor noise (Fig. 10). We therefore render a more realistic *gel-rendered* image on top of the same signal, without re-simulating: we rebuild a bright silver gel base from the optical image's luminance, overlay a dense, *markerless* field of fine colored speckles, and *advect* that field with the contact, tangentially by the per-vertex surface displacement (the same field that drives the marker motion) and normally by the depth-indentation gradient, which pushes the speckles outward as the gel bulges around the indenter; a faint depth shade and elastomer grain complete the look. Contact then reads out of a *deforming textured surface* rather than a color code. Trained and evaluated on this rendering, $\mathcal{N}_0$-TWAM stays close to the clean field rather than dropping sharply: 82.4% against 84.5% on the eight UniVTAC tasks, and 58.5% against 63.8% (single-arm) and 39.3% against 42.3% (dual-arm) on NeoSim (Table 6; the per-task breakdown is in Appendix B). The realistic appearance therefore costs little accuracy in-domain, even though it replaces the clean synthetic blob with sensor noise, grain, and a deforming textured surface. Because it shares the visual domain of a real vision-based sensor, the gel-rendered image is also a tactile stream a pretrained force encoder can read, which we examine below.

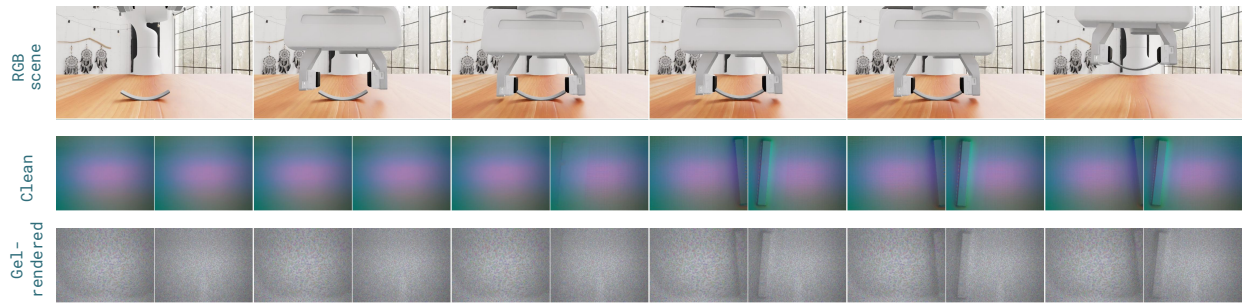

Figure 10 Two renderings of the same tactile stream. One observed rollout: the top row is the RGB scene, shared by both renderings; below it the same contact appears as the simulator's smooth *clean* field and as our *gel-rendered* rendering, a silvery, speckled gel with grain noise and depth shading that matches the look of a real vision-based sensor. Only the tactile rendering differs.

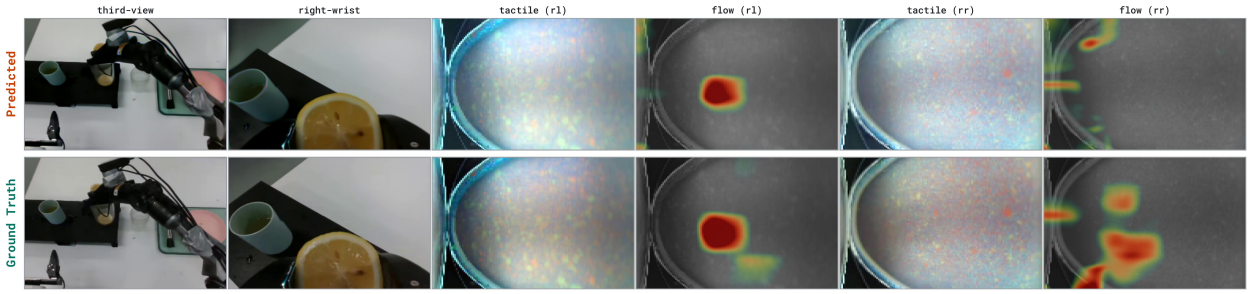

Figure 11 Predicting future touch (lemon-tea grasp). Top row: the model's *predicted* future; bottom row: the *ground truth*. From left to right, the third-person and right-wrist RGB views, then the tactile image and the contact-flow map for each of the two gripper fingers (*rl*, *rr*). The predicted tactile and flow match the observed sensor, so foresight extends to contact, not only to the visual scene.

Table 6 Realism of the simulated tactile, and the pretrained NeoForce force encoder on it. Average success (%) for the simulator's clean field, our gel-rendered image, and the gel-rendered image read through NeoForce, on UniVTAC (eight tasks) and NeoSim (four single-arm and eight dual-arm tasks). Per-task numbers are in Appendix B.

Tactile	UniVTAC	NeoSim (single)	NeoSim (dual)
Clean (simulator)	84.5	63.8	42.3
Gel-rendered	82.4	58.5	39.3
Gel-rendered + NeoForce	**88.1**	**64.8**	/

Pretrained force encoder on the gel-rendered tactile. NeoForce is pretrained on real vision-based tactile images, the gel-particle appearance our gel-rendered render imitates. Because the two share this domain, reading the gel-rendered tactile through NeoForce, instead of the observed-tactile encoder trained from scratch, raises success from 82.4% to 88.1% on the eight UniVTAC tasks and from 58.5% to 64.8% on the four single-arm NeoSim tasks (Table 6; per-task numbers in Appendix B). The gains concentrate on tasks whose success turns on a weight-bearing or shear cue that a force representation reads well: Put Bottle in Shelf (82 → 96), Insert HDMI (63 → 75), and Pull-out Key (74 → 83) on UniVTAC, and Grasp Chip (82 → 92) and Pour Ball (55 → 68) on NeoSim, while the sharp-contact insertion tasks are already near ceiling and change little. We still report the clean field in the main results, where every method uses the same simulated tactile; the gel-rendered + NeoForce number is a sim-to-real signal, showing that once the simulated touch looks real, a force encoder pretrained on real touch transfers into it.

Tactile foresight. A direct test of whether the model truly *predicts* touch is to compare its predicted future tactile against the ground truth (Fig. 11). On the lemon-tea grasp, the predicted tactile images and their

contact-flow maps track the ground-truth sensor closely for both fingers: the model anticipates where and how contact forms, not only the future scene. This is the qualitative signal we expect if predicting *touch*, rather than appearance alone, is what drives the downstream gains.

5 Related Work

Vision-language-action models. Vision-language-action (VLA) models cast manipulation as conditional action generation from observations and language. RT-style and open generalist policies [7–9, 23, 38, 97] show that large-scale vision-language pretraining transfers semantic knowledge into robot control. Recent systems extend this recipe through real-time asynchronous execution and much larger cross-embodiment corpora [12, 72, 74], while embodied backbones and unified policies add action-centric reasoning and visual or latent foresight [11, 54, 63, 67]. Yet these foundation-scale systems still center on vision, language, and proprioception. They are typically measured on unified sim-and-real benchmarks such as VLABench [89], LIBERO-Plus [22], and RoboDojo [17] that span generalization, memory, precision, and long-horizon skills, but not the contact-rich, tactile-dependent tasks $\mathcal{N}_0$-TWAM targets. In contact-rich manipulation, the deciding variable may instead be fingertip pressure, incipient slip, or imminent collision; even predictive variants model visual or latent futures rather than an explicit tactile trajectory consumed by the action expert. $\mathcal{N}_0$-TWAM targets this gap.

Tactile perception for manipulation. Tactile perception supplies the contact signals vision misses. Tactile-aware VLA systems [29, 36, 83, 88, 90] align tactile observations with language and action [29, 36], inject force feedback into the action expert [83], specialize for insertion and other contact-heavy tasks [88], or steer a pretrained visuomotor policy at inference with a touch-derived feasibility signal [90]. Force-aware policies fuse RGB with high-rate force/torque streams, learn task-dependent compliance, or add torque adapters to a VLA decoder [30, 33, 41, 78, 92], and visuo-tactile policies show that touch improves fine-grained manipulation and stays robust when vision is ambiguous or occluded [6, 18, 35, 71, 76, 93]. In this direct-policy family touch is largely *reactive* [57]: it enters as an immediate observation for the next action rather than a contact trajectory the model predicts before acting. A more recent line makes touch *predictive*: DreamTacVLA fine-tunes a tactile world model so actions condition on both real and predicted tactile consequences [80], predictive visuo-tactile policies autoregress future contact or generate force-domain action trajectories [31, 73], and visuo-tactile world models and tactile action models study contact-aware prediction for planning and closed-loop control [32, 52, 64, 70, 85, 87, 91, 94]. Against this prior art the question $\mathcal{N}_0$-TWAM addresses is not whether predicted touch helps, but how to keep it in the action loop at scale while preserving a pretrained visual world model.

World models for robotics. World models learn a predictive model of the environment and roll out future states conditioned on text [65], driving commands [34], or robot actions [69]; trained on large video corpora, world foundation models [58] generate high-fidelity future observations, recover action-controllable dynamics from unlabelled video [10], or predict in a latent space rather than pixels [4]. Rather than map the current observation directly to an action, a complementary line first *predicts* a future and then derives the action from it, from model-based agents that optimize a policy against predicted rollouts [3, 25–28] to video-first policies that plan through predicted visual futures [20, 21, 40, 49]. World-action models (WAMs) couple the two stages in a single model [13, 24, 45, 47, 68, 82, 96], sharing or repurposing visual latent spaces for action [5, 14, 39, 66] under various temporal abstractions [42, 46, 53]. To give each modality its own capacity while keeping a shared reasoning path, MoT [50] decouples per-modality feed-forward, attention, and normalization but preserves a shared global attention [55]; LingBot-VA, which $\mathcal{N}_0$-TWAM builds on, applies an MoT WAM over vision and action tokens [45], and causal video training improves long-horizon rollouts [37, 75]. Recent WAMs differ chiefly in how they pay for future prediction at test time: FastWAM collapses the video branch into a single-pass encoder [86], Metis lets the action expert bypass future video generation [44], and Efficient-WAM prunes and shortens the video rollout [43], with related distillation and short-cutting elsewhere [2, 62, 77]; across this line efficiency is usually bought by weakening or bypassing the predicted future, and the predicted modality is pixels. $\mathcal{N}_0$-TWAM occupies the tactile version of this space: it orders the video, tactile, and action experts into a cascade so predicted touch is an intermediate

prediction the action expert consumes rather than a side signal or attention bias [64, 70], and it allocates width asymmetrically to keep full pretrained capacity in the visual expert while training slimmer tactile and action experts. The same asymmetry keeps streaming inference cheap without pruning the predicted future away.

6 Conclusion

We presented $\mathcal{N}_0$-TWAM, a tactile-native world-model policy that brings touch into the predicted future of a video world-action model. By organizing a pretrained video-diffusion backbone into three per-modality experts tied by a single shared self-attention, and ordering them with a frame-id causal cascade, $\mathcal{N}_0$-TWAM predicts the near-future scene and the near-future contact and then reads action out of both. Tactile is given a dual role (a denoised foresight target and an observed reading), while an asymmetric design keeps the model at 7.2B and streaming inference cheap, since each action-denoising step re-runs only the lightweight action expert. Crucially, touch is not an add-on evaluated on a handful of tasks: $\mathcal{N}_0$-TWAM is pre-trained at scale, on a large self-collected real-robot data with synchronized tactile, so that a shared tactile representation is learned jointly with vision rather than fitted per task. Across contact-rich benchmarks it is the strongest method overall, in simulation (84.5 on UniVTAC, 49.4 on NeoSim) and on real robots (46.3% average, ahead of every vision-language-action and vision-only world-action baseline), with ablations confirming that both the predicted and the observed tactile pathways contribute.

Future work. We see three main directions. First, *faster inference*: streaming decoding of the one-directional cascade can be accelerated further, so the policy runs at higher control rates. Second, *a longer predicted horizon*: extending the vision and tactile prediction window would let the model anticipate contact events further ahead. Third, *broader sensor coverage*: training on more types and varieties of tactile sensors would widen the range of embodiments the learned tactile representation transfers to.

Contributors

Pre-Training. Li Kang, Xiufeng Song, Yanjun Li, Rui Li, Shunlin Lu, Yiran Qin.
Post-Training (Simulation). Yifan Wang, Li Kang, Zipei Ma, Bruno N.Y. Chen, Heng Zhou.
Post-Training (Real Robot). Li Kang, Yanjun Li, Zipei Ma, Shengqi Xu, Boyu Mi, Bruno N.Y. Chen, Heng Zhou, Ren Jia, Silong Dai, Jiongwei Lu.
Data Processing. Li Kang, Boyu Mi, Rui Li, Xiufeng Song, Yifan Wang, Yutao Fan, Longjie Su, Zhemeng Zhang, Xin Wang, Tianyu Yang, Wenjie Zhou.
Academic Supervision. Ziyi Ye, Guoxiang Dong, Xiaosong Jia, Wenming Chen.
Project Lead. Yiran Qin, Shunlin Lu, Shihao Zhao, Daoguo Dong, Zuxuan Wu.

References

[1] Michael Ahn, Anthony Brohan, Noah Brown, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

[2] Arman Akbari, Ci Zhang, Arash Akbari, Lin Zhao, Yixiao Chen, Weiwei Chen, Xuan Zhang, Geng Yuan, and Yanzhi Wang. Flash-WAM: Modality-aware distillation for world action models. *arXiv preprint arXiv:2606.05254*, 2026.

[3] Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. *NeurIPS*, 2024.

[4] Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, et al. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

[5] 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. *arXiv preprint arXiv:2512.13030*, 2025.

[6] Jianxin Bi, Kevin Yuchen Ma, Ce Hao, Mike Zheng Shou, and Harold Soh. Vla-touch: Enhancing vision-language-action models with dual-level tactile feedback. *arXiv preprint arXiv:2507.17294*, 2025.

[7] Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert 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. *CoRL*, 2025.

[8] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. *RSS*, 2025.

[9] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael S. Ryoo, Grecia Salazar, Pannag R. Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan H. Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-1: Robotics transformer for real-world control at scale. *RSS*, 2023.

[10] Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Bechtle, Feryal Behbahani, Stephanie Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments. *ICML*, 2024.

[11] Junhao Cai, Zetao Cai, Jiafei Cao, Yilun Chen, Zeyu He, Lei Jiang, Hang Li, Hengjie Li, Yang Li, Yufei Liu, et al. InternVLA-A1: Unifying understanding, generation and action for robotic manipulation. *arXiv preprint arXiv:2601.02456*, 2026.

[12] Rui Cai, Jun Guo, Xinze He, Piaopiao Jin, Jie Li, Bingxuan Lin, Futeng Liu, Wei Liu, Fei Ma, Kun Ma, et al. Xiaomi-Robotics-0: An open-sourced vision-language-action model with real-time execution. *arXiv preprint arXiv:2602.12684*, 2026.

[13] Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, Deli Zhao, and Hao Chen. WorldVLA: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025.

[14] Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, Hanbo Zhang, and Minzhao Zhu. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024.

[15] Baijun Chen, Weijie Wan, Tianxing Chen, Xianda Guo, Congsheng Xu, Yuanyang Qi, Haojie Zhang, Longyan Wu, Tianling Xu, Zixuan Li, et al. UniVTAC: A unified simulation platform for visuo-tactile manipulation data generation, learning, and benchmarking. *arXiv preprint arXiv:2602.10093*, 2026.

[16] Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *NeurIPS*, 2024.

[17] Tianxing Chen, Yue Chen, Zixuan Li, Junyuan Tang, Kailun Su, Weijie Wan, Baijun Chen, Haoran Lu, Haowen Yan, Honghao Su, et al. RoboDojo: A unified sim-and-real benchmark for comprehensive evaluation of generalist robot manipulation policies. *arXiv preprint arXiv:2607.04434*, 2026.

[18] Zhengxue Cheng, Yiqian Zhang, Anni Tang, Keyu Wang, Wenkang Zhang, Haoyu Li, Hengdi Zhang, and Li Song. Omnivtla: Vision-tactile-language-action models with semantic-aligned tactile sensing. *RA-L*, 2025.

[19] Google DeepMind. Gemini 3.5 flash. https://deepmind.google/models/gemini/flash/, 2025.

[20] 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. *NeurIPS*, 2023.

[21] Yilun Du, Mengjiao Yang, Pete Florence, Fei Xia, Ayzaan Wahid, Brian Ichter, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B. Tenenbaum, Leslie Kaelbling, Andy Zeng, and Jonathan Tompson. Video language planning. *ICLR*, 2024.

[22] Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. LIBERO-plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

[23] Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Quan Vuong, Ted Xiao, Pannag R. Sanketi, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. *RSS*, 2024.

[24] Yanjiang Guo, Yucheng Hu, Jianke Zhang, Yen-Jen Wang, Xiaoyu Chen, Chaochao Lu, and Jianyu Chen. Prediction with action: Visual policy learning via joint denoising process. *NeurIPS*, 2024.

[25] David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. *NeurIPS*, 2018.

[26] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. *ICML*, 2019.

[27] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *ICLR*, 2020.

[28] Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. *ICLR*, 2021.

[29] Peng Hao, Chaofan Zhang, Dingzhe Li, Xiaoge Cao, Xiaoshuai Hao, Shaowei Cui, and Shuo Wang. TLA: Tactile-language-action model for contact-rich manipulation. *arXiv preprint arXiv:2503.08548*, 2025.

[30] Zihao He, Hongjie Fang, Jingjing Chen, Hao-Shu Fang, and Cewu Lu. Foar: Force-aware reactive policy for contact-rich robotic manipulation. *RA-L*, 2024.

[31] Liang Heng, Haoran Geng, Kaifeng Zhang, Pieter Abbeel, and Jitendra Malik. Vitacformer: Learning cross-modal representation for visuo-tactile dexterous manipulation. *arXiv preprint arXiv:2506.15953*, 2025.

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

[33] 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. *arXiv preprint arXiv:2410.09309*, 2024.

[34] Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. GAIA-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023.

[35] Binghao Huang, Yixuan Wang, Xinyi Yang, Yiyue Luo, and Yunzhu Li. 3D-ViTac: Learning fine-grained manipulation with visuo-tactile sensing. *CoRL*, 2025.

[36] Jialei Huang, Shuo Wang, Fanqi Lin, Yihang Hu, Chuan Wen, and Yang Gao. Tactile-VLA: Unlocking vision-language-action model's physical knowledge for tactile generalization. *arXiv preprint arXiv:2507.09160*, 2025.

[37] Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *NeurIPS*, 2025.

[38] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P. Foster, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. *CoRL*, 2025.

[39] 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. *arXiv preprint arXiv:2601.16163*, 2026.

[40] Po-Chen Ko, Jiayuan Mao, Yilun Du, Shao-Hua Sun, and Joshua B. Tenenbaum. Learning to act from actionless videos through dense correspondences. *ICLR*, 2024.

[41] Geonhyup Lee, Yeongjin Lee, Kangmin Kim, Seongju Lee, Sangjun Noh, Seunghyeok Back, and Kyoobin Lee. Manipforce: Force-guided policy learning with frequency-aware representation for contact-rich manipulation. *arXiv preprint arXiv:2509.19047*, 2025.

[42] Hao Li, Shuai Yang, Yilun Chen, Xinyi Chen, Xiaoda Yang, Yang Tian, Hanqing Wang, Tai Wang, Dahua Lin, Feng Zhao, and Jiangmiao Pang. Cronusvla: Towards efficient and robust manipulation via multi-frame vision-language-action modeling. *arXiv preprint arXiv:2506.19816*, 2025.

[43] Jiajun Li, Tiecheng Guo, Yifan Ye, Rongyu Zhang, Xiaowei Chi, Qianpu Sun, Ying Li, Yunfan Lou, Yan Huang, Zhihe Lu, Meng Guo, and Shanghang Zhang. Efficient-WAM: A 1B-parameter world-action model with low-cost future imagination. *arXiv preprint arXiv:2606.10040*, 2026.

[44] Jingyu Li, Zhe Liu, Dongnan Hu, Junjie Wu, Zipei Ma, Wenxiao Wu, Chao Han, Zhihui Hao, Zhikang Liu, Kun Zhan, Jiankang Deng, Xiatian Zhu, and Li Zhang. Metis: A generalizable and efficient world-action model for autonomous driving and urban navigation. *arXiv preprint arXiv:2606.15869*, 2026.

[45] Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Zhangluyao, Mingrui Yu, Zelin Gao, Nan Xue, Boyu Zhou, Xing Zhu, Mingyu Ding, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026.

[46] Shalfun Li, Victor Yao, Charles Yang, Truth Qu, Regis Cheng, Ryan Yu, Howard Lu, Newton Von, Vincent Chen, Yohann Tang, Maeve Zhang, Ellie Ma, Gody Li, Sage Yang, Lorien Shu, J. W. Gao, Ethan Chen, Colin Ye, Yu Sun, Elise Mon, PS Zhang, Neo Li, Lily Li, James Wang, Ping Yang, Chris Pan, Lucy Liang, Hang Su, Roy Gan, Hao Wang, and Qian Wang. Wall-wm: Carving world action modeling at the event joints. *arXiv preprint arXiv:2606.01955*, 2026.

[47] Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. *RSS*, 2025.

[48] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. *arXiv preprint arXiv:2209.07753*, 2022.

[49] Junbang Liang, Ruoshi Liu, Ege Ozguroglu, Sruthi Sudhakar, Achal Dave, Pavel Tokmakov, Shuran Song, and Carl Vondrick. Dreamitate: Real-world visuomotor policy learning via video generation. *CoRL*, 2024.

[50] Weixin Liang, Lili Yu, Liang Luo, Srinivasan Iyer, Ning Dong, Chunting Zhou, Gargi Ghosh, Mike Lewis, Wen-tau Yih, Luke Zettlemoyer, and Xi Victoria Lin. Mixture-of-transformers: A sparse and scalable architecture for multi-modal foundation models. *TMLR*, 2025.

[51] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *ICLR*, 2023.

[52] 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.

[53] Hao Luo, Wanpeng Zhang, Yicheng Feng, Sipeng Zheng, Haiweng Xu, Chaoyi Xu, Ziheng Xi, Yuhui Fu, and Zongqing Lu. Being-h0.7: A latent world-action model from egocentric videos. *arXiv preprint arXiv:2605.00078*, 2026.

[54] Haoxiang Ma, Junhao Cai, Xiaoxu Xu, Hao Li, Yuyin Yang, Yang Tian, Jiafei Cao, Hongrui Zhu, Zherui Qiu, et al. Internvla-a1.5: Unifying understanding, latent foresight, and action for compositional generalization. *arXiv preprint arXiv:2607.04988*, 2026.

[55] Shuailei Ma, Jiaqi Liao, Xinyang Wang, Jingjing Wang, Chaoran Feng, Zijing Hu, Chong Bao, Zichen Xi, Yuqi Gan, Weisen Wang, Yanhong Zeng, Qin Zhao, Zifan Shi, Wei Wu, Hao Ouyang, Qiuyu Wang, Shangzhan Zhang, Jiahao Shao, Yipengjing Sun, Liangxiao Hu, Lunke Pan, Nan Xue, Kecheng Zheng, Yinghao Xu, Xing Zhu, Yujun Shen, and Ka Leong Cheng. Scaling mixture-of-experts video pretraining for embodied intelligence. *arXiv preprint arXiv:2607.07675*, 2026.

[56] NeoteAI Team and Fudan TEAI Team. $\mathcal{N}_0$-foundation: Towards the age of tactile intelligence. https://research.neoteai.com/n0-foundation/, 2026.

[57] Dantong Niu, Zhuoyang Liu, Zekai Wang, ..., Linxi Fan, and Trevor Darrell. T-Rex: Tactile-reactive dexterous manipulation. *arXiv preprint arXiv:2606.17055*, 2026.

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

[59] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, et al. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[60] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. *CoRL*, 2022.

[61] Zilin Si and Wenzhen Yuan. Taxim: An example-based simulation model for GelSight tactile sensors. *RA-L*, 2022.

[62] Wenxuan Song, Jiayi Chen, Shuai Chen, Jingbo Wang, Pengxiang Ding, Han Zhao, Yikai Qin, Xinhu Zheng, Donglin Wang, Yan Wang, and Haoang Li. Fast-dvla: Accelerating discrete diffusion VLA to real-time performance. *arXiv preprint arXiv:2603.25661*, 2026.

[63] Tencent Robotics X, HY Vision Team, Xumin Yu, Zuyan Liu, Ziyi Wang, He Zhang, Yongming Rao, Fangfu Liu, Yani Zhang, Ruowen Zhao, Oran Wang, Yves Liang, Haitao Lin, Minghui Wang, Yubo Dong, Kevin Cheng, Bolin Ni, Rui Huang, Han Hu, Zhengyou Zhang, Linus, and Shunyu Yao. Hy-embodied-0.5: Embodied foundation models for real-world agents. *arXiv preprint arXiv:2604.07430*, 2026.

[64] 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.

[65] Wan Team, Ang Wang, Baole Ai, Bin Wen, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[66] Junke Wang, Qihang Zhang, Shuai Yang, Yiming Luo, Yujun Shen, Zuxuan Wu, Yu-Gang Jiang, and Yinghao Xu. Repwam: World action modeling with representation visual-action tokenizers. *arXiv preprint arXiv:2606.13674*, 2026.

[67] Ziyi Wang, Xumin Yu, Yongming Rao, Yonggen Ling, Yunheng Li, Oran Wang, Mingqi Gao, Yuchen Zhou, Yves Liang, Zuyan Liu, Yani Zhang, Rui Huang, Xiaoran Xu, Bowen Yuan, Yifu Yuan, Xu Tan, He Zhang, Yufei Huang, Shenghao Zhang, Hongsheng Wu, Han Hu, and Zhengyou Zhang. Hy-embodied-vlm-1.0: Efficient physical-world agents. *arXiv preprint arXiv:2607.12894*, 2026.

[68] Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. *ICLR*, 2024.

[69] Jialong Wu, Shaofeng Yin, Ningya Feng, Xu He, Dong Li, Jianye Hao, and Mingsheng Long. iVideoGPT: Interactive VideoGPTs are scalable world models. *NeurIPS*, 2024.

[70] Siyu Wu, Linjing You, Junjie Zhu, Yaozu Liu, Changhao Zhang, Jian Liu, Weiqiang Wang, Qi Li, Jituo Li, and Hengshuang Zhao. Tactile-WAM: Touch-aware world action model with tactile asymmetric attention. *arXiv preprint arXiv:2606.26663*, 2026.

[71] Tianhao Wu, Jinzhou Li, Jiyao Zhang, Mingdong Wu, and Hao Dong. Canonical representation and force-based pretraining of 3d tactile for dexterous visuo-tactile policy learning. *ICRA*, 2024.

[72] Wei Wu, Fangjing Wang, Fan Lu, He Sun, Shi Liu, Yunnan Wang, Yibin Yan, Yong Wang, Shuailei Ma, Xinyang Wang, Yibin Liu, Shuai Yang, Tianxiang Zhou, Kejia Zhang, Lei Zhou, Cheng Su, Nan Xue, Bin Tan, Han Zhang, Youchao Zhang, Fei Liao, Xing Zhu, Yujun Shen, and Kecheng Zheng. From foundation to application: Improving vla models in practice. *arXiv preprint arXiv:2607.06403*, 2026.

[73] Yansong Wu, Zongxie Chen, Fan Wu, Lingyun Chen, Liding Zhang, Zhenshan Bing, Abdalla Swikir, Sami Haddadin, and Alois Knoll. Tacdiffusion: Force-domain diffusion policy for precise tactile manipulation. *ICRA*, 2024.

[74] Xiaomi Robotics Team, Jun Guo, Piaopiao Jin, Jason Li, Peiyan Li, Yingyan Li, Futeng Liu, Wanli Peng, Optimus Qin, Yifei Su, Nan Sun, Qiao Sun, Runze Suo, Heyun Wang, Yunhong Wang, Rujie Wu, Caoyu Xia, Lina Zhang, Jack Zhao, Guoliang Chen, Wenlong Chen, Xinze He, Bin Li, Qing Li, Zhuorong Li, Heng Qu, Wenxuan Song, Diyun Xiang, Yifan Xie, Peiran Xu, Hangjun Ye, Wen Ye, Han Zhao, and Quanyun Zhou. Xiaomi-robotics-1: Scaling vision-language-action models with over 100k hours of real-world trajectories. *arXiv preprint arXiv:2607.15330*, 2026.

[75] Gangwei Xu, Qihang Zhang, Jiaming Zhou, Xing Zhu, Yujun Shen, Xin Yang, and Yinghao Xu. Next forcing: Causal world modeling with multi-chunk prediction. *arXiv preprint arXiv:2606.11187*, 2026.

[76] Shengqi Xu, Guojin Zhong, Yang Liu, Fanjie Wang, Hu Luo, Hanyu Zhou, Weiyao Zhang, Ziyi Ye, Zuxuan Wu, and Yu-Gang Jiang. Seeing touch from motion: A unified modality-aware visuo-tactile policy with tactile motion correlation. *ECCV*, 2026.

[77] Haodong Yan, Zhide Zhong, Jiaguan Zhu, Junjie He, Weilin Yuan, Wenxuan Song, Xin Gong, Yingjie Cai, Guanyi Zhao, Xu Yan, Bingbing Liu, Ying-Cong Chen, and Haoang Li. S-VAM: Shortcut video-action model by self-distilling geometric and semantic foresight. *arXiv preprint arXiv:2603.16195*, 2026.

[78] Taozheng Yang, Ya Jing, Hongtao Wu, Jiafeng Xu, Kuankuan Sima, Guangzeng Chen, Qie Sima, and Tao Kong. Moma-force: Visual-force imitation for real-world mobile manipulation. *IROS*, 2023.

[79] Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, et al. GigaWorld-Policy: An efficient action-centered world-action model. *arXiv preprint arXiv:2603.17240*, 2026.

[80] Guo Ye, Zexi Zhang, Xu Zhao, Shang Wu, Haoran Lu, Shihan Lu, and Han Liu. Learning to feel the future: DreamTacVLA for contact-rich manipulation. *arXiv preprint arXiv:2512.23864*, 2025.

[81] Jinhui Ye, Ning Gao, Senqiao Yang, Jinliang Zheng, Zixuan Wang, Yuxin Chen, Pengguang Chen, Yilun Chen, Shu Liu, and Jiaya Jia. StarVLA-α: Reducing complexity in vision-language-action systems. *arXiv preprint arXiv:2604.11757*, 2026.

[82] Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

[83] 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. *NeurIPS*, 2025.

[84] Chengbo Yuan, Zicheng Zhang, ..., Kaifeng Zhang, and Yang Gao. FTP-1: A generalist foundation tactile policy across tactile sensors for contact-rich manipulation. *arXiv preprint arXiv:2606.13102*, 2026.

[85] Haoran Yuan, Weigang Yi, Zhenyu Zhang, Wendi Chen, Yuchen Mo, Jiashi Yin, Xinzhuo Li, Xiangyu Zeng, Chuan Wen, Cewu Lu, Katherine Driggs-Campbell, and Ismini Lourentzou. VTAM: Video-tactile-action models for complex physical interaction beyond VLAs. *arXiv preprint arXiv:2603.23481*, 2026.

[86] 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.

[87] Yujie Zang, Yuhang Zheng, Xian Nie, Yupeng Zheng, Shuai Tian, Songen Gu, Chen Gao, Zining Wang, Shuicheng Yan, and Wenchao Ding. TacForeSight: Force-guided tactile world model for contact-rich manipulation. *arXiv preprint arXiv:2606.11184*, 2026.

[88] Chaofan Zhang, Peng Hao, Xiaoge Cao, Xiaoshuai Hao, Shaowei Cui, and Shuo Wang. VTLA: Vision-tactile-language-action model with preference learning for insertion manipulation. *Biomimetic Intelligence and Robotics*, 2026.

[89] Shiduo Zhang, Zhe Xu, Peiju Liu, Xiaopeng Yu, Yuan Li, Qinghui Gao, Zhaoye Fei, Zhangyue Yin, Zuxuan Wu, Yu-Gang Jiang, and Xipeng Qiu. VLABench: A large-scale benchmark for language-conditioned robotics manipulation with long-horizon reasoning tasks. *arXiv preprint arXiv:2412.18194*, 2024.

[90] Zhemeng Zhang, Jiahua Ma, Xincheng Yang, Xin Wen, Yuzhi Zhang, Boyan Li, Yiran Qin, Jin Liu, Can Zhao, Li Kang, Haoqin Hong, Zhenfei Yin, Philip Torr, Hao Su, Ruimao Zhang, and Daolin Ma. Touchguide: Inference-time steering of visuomotor policies via touch guidance. *arXiv preprint arXiv:2601.20239*, 2026.

[91] Zhiyuan Zhang, Pokuang Zhou, Kaidi Zhang, Adeesh Desai, Temitope Amosa, Davood Soleymanzadeh, Jiuzhou Lei, Minghui Zheng, and Yu She. ContactWorld: What matters in vision-tactile world models for contact-rich manipulation. *arXiv preprint arXiv:2606.13877*, 2026.

[92] Zongzheng Zhang, Haobo Xu, Zhuo Yang, Chenghao Yue, Zehao Lin, Huan-ang Gao, Ziwei Wang, and Hao Zhao. Ta-vla: Elucidating the design space of torque-aware vision-language-action models. *CoRL*, 2025.

[93] Zifan Zhao, Siddhant Haldar, Jinda Cui, Lerrel Pinto, and Raunaq Bhirangi. Touch begins where vision ends: Generalizable policies for contact-rich manipulation. *arXiv preprint arXiv:2506.13762*, 2025.

[94] Yuhang Zheng, Songen Gu, Weize Li, Yupeng Zheng, Yujie Zang, Shuai Tian, Xiang Li, 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.

[95] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. *CVPR*, 2019.

[96] Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. *RSS*, 2025.

[97] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: Vision-language-action models transfer web knowledge to robotic control. *CoRL*, 2023.

A Per-task ablation results

Table 7 gives the per-task UniVTAC success rates behind the averages in Figure 9, for the full model and the three ablations, and Table 8 gives the twelve-task NeoSim ablation (four single-arm and eight dual-arm).

Table 7 Per-task UniVTAC ablation of $\mathcal{N}_0$-TWAM (success rate %) over the eight tasks.

Variant	Insert HDMI	Insert Hole	Insert Tube	Grasp Classify	Lift Can	Lift Bottle	Pull-out Key	Put Bottle in Shelf	Avg.
$\mathcal{N}_0$-TWAM (full)	68	99	98	94	93	58	79	87	84.5
20% pre-train	52	78	96	95	80	51	25	46	65.4
w/o predicted	73	97	96	93	91	16	86	22	71.8
w/o observed	68	93	99	58	68	0	86	92	70.5

Table 8 Per-task NeoSim ablation of $\mathcal{N}_0$-TWAM (success rate %) over the twelve tasks (four single-arm and eight dual-arm). The 20% pre-training variant was run on UniVTAC only.

	Single-arm				Dual-arm								
Variant	Insert USB	Grasp Chip	Unplug & Plug Charger	Pour Ball	Bowl Unstack	Cup Stack	Insert Screw	Place Gears	Cup Handover	Plate Stack	Bowl Stack	Cup Unstack	Avg.
$\mathcal{N}_0$-TWAM (full)	100	92	0	63	72	12	29	0	14	98	97	16	49.4
w/o predicted	82	61	0	38	18	22	6	0	65	96	97	8	41.1
w/o observed	88	24	0	17	32	38	10	0	30	15	96	5	29.6

B Per-task tactile-realism results

Tables 9 and 10 give the per-task success behind Table 6: $\mathcal{N}_0$-TWAM on the simulator's clean field, on our gel-rendered image, and on the gel-rendered image read through the pretrained NeoForce force encoder. The clean and gel-rendered results are close on most tasks, so the realistic appearance costs little accuracy in-domain. Reading the gel-rendered image through NeoForce improves it further, most on the tasks with a clear weight-bearing or shear cue (Put Bottle in Shelf 82→96, Insert HDMI 63→75, Pull-out Key 74→83 on UniVTAC; Grasp Chip 82→92, Pour Ball 55→68 on NeoSim).

Table 9 Per-task success rate (%) on the eight UniVTAC tasks for the clean field, the gel-rendered image, and the gel-rendered image read through NeoForce.

Tactile	Insert HDMI	Insert Hole	Insert Tube	Grasp Classify	Lift Can	Lift Bottle	Pull-out Key	Put Bottle in Shelf	Avg.
Clean	68	99	98	94	93	58	79	87	84.5
Gel-rendered	63	98	99	97	94	52	74	82	82.4
Gel-rendered + NeoForce	75	98	100	96	99	58	83	96	88.1

Table 10 Per-task success rate (%) on the twelve NeoSim tasks (four single-arm and eight dual-arm) for the clean field, the gel-rendered image, and the gel-rendered image read through NeoForce.

	Single-arm				Dual-arm								
Tactile	Insert USB	Grasp Chip	Unplug & Plug Charger	Pour Ball	Bowl Unstack	Cup Stack	Insert Screw	Place Gears	Cup Handover	Plate Stack	Bowl Stack	Cup Unstack	Avg.
Clean	100	92	0	63	72	12	29	0	14	98	97	16	49.4
Gel-rendered	97	82	0	55	74	20	17	0	12	96	83	12	45.7
Gel-rendered + NeoForce	99	92	0	68	/	/	/	/	/	/	/	/	/

C Per-modality training dynamics

We consider two pre-training data recipes: one that trains on real-robot teleoperation data only, and one that additionally mixes in the hand-collected UMI data and then continues on the pure real-robot teleoperation data. Figure 12 shows the latter over the first 18k steps (the full run is 30,000 steps). Because the MoT keeps a separate expert per modality, we can log the flow-matching loss separately for the video, action, and tactile streams. Before the +UMI marker all three fall and stabilize. When the UMI data is mixed in (about 60% of the batch) the modalities respond differently: the *video* loss rises at the transition and settles at a somewhat higher value than before, since the UMI footage is a new visual distribution the video expert must now also fit; the *tactile* loss barely moves, because the tactile signal is similar across the two data sources; and the *action* loss shows only a small step.

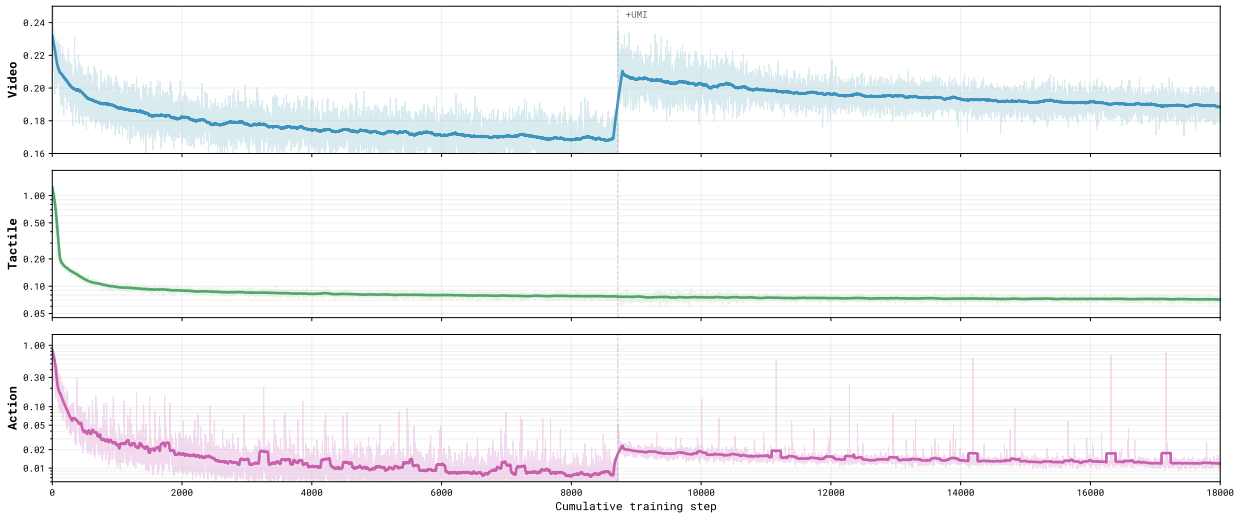

Figure 12 Per-modality pre-training loss (first 18k steps). Video, action, and tactile flow-matching losses of the MoT backbone; the +UMI marker is where the hand-collected UMI data is mixed in. The video loss rises and converges higher afterward, while the tactile loss is essentially unchanged.