TACDYN-WAM: LEARNING IMPLICIT TACTILE DYNAMICS IN A HETEROGENEOUS VISUO-TACTILE WORLD ACTION MODEL

Anonymous authors
Paper under double-blind review

ABSTRACT

World action models improve robotic manipulation by conditioning actions on predicted futures, yet existing tactile variants largely inherit video-generation pipelines that reconstruct future tactile observations through iterative denoising. Such prediction can become unreliable under deployment drift: small changes in contact position or force may substantially alter tactile pixels even when the underlying contact evolution remains predictable. We introduce TacDyn-WAM, a heterogeneous visuo-tactile world action model that predicts implicit tactile dynamics rather than reconstructing future tactile observations. It learns TacRep, a dynamics-aware tactile target space trained through masked spatio-temporal prediction on tactile clips and regularized by relational structure distillation. A visual expert and an Implicit Tactile Dynamics Expert predict future visual and tactile representations in separate target spaces while interacting through joint attention; the tactile expert predicts future representations and their changes at multiple horizons in a single forward pass, and a read-only tactile memory supplies the current tactile state. On UniVTAC, TacDyn-WAM achieves an average success rate of 81.5% using only the provided demonstrations, reaching state-of-the-art-level performance and remaining competitive with models pretrained on large-scale visuo-tactile trajectories. Ablations confirm the benefits of both tactile pathways and TacRep over pixel-reconstruction and static alternatives. On five real-robot tasks, TacDyn-WAM reaches 71.0% average success, and modest-scale pretraining raises it to 85.0%, further validating our method.

1 INTRODUCTION

World models forecast how observations evolve, while world action models use this foresight for policy learning, allowing actions to depend on predicted futures rather than current observations alone (Cai et al., 2026a; Li et al., 2026; Yuan et al., 2026c; Ye et al., 2026a). Yet in existing world action models, this foresight comes primarily from vision. In contact-rich or precision manipulation tasks, such as insertion and alignment under occlusion, cameras cannot fully resolve contact states, local geometry, or interaction forces. Early efforts incorporated tactile sensing into VLA models, giving rise to vision-tactile-language-action (VTLA) models that encode tactile observations alongside vision and language (Huang et al., 2025; Zhang et al., 2026a; Bi et al., 2026b; Cheng et al., 2026). By understanding the current tactile state, they improve contact-rich manipulation over vision-only policies. However, they do not predict how this state will evolve, leaving policies reactive to tactile feedback. Recent tactile world action models close this gap by predicting future tactile observations as part of the world model (NeoteAI Team et al., 2026a; Lou et al., 2026; Yuan et al., 2026b; Wu et al., 2026). These models have achieved promising results, with some further benefiting from large-scale visuo-tactile pretraining. Despite this progress, a core design question remains: what should tactile world models predict?

Most existing methods simply reuse video-generation pipelines: the same VAE encodes camera and tactile frames, while video backbones iteratively denoise future tactile latents. This slows inference and ties prediction to pixel reconstruction, which is poorly suited to tactile observations. Expert demonstrations cover only tactile images near correct contacts, whereas closed-loop errors may cause policies to drift from them. This drift may leave camera views nearly unchanged, yet even

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

slight changes in contact position or force can substantially alter tactile images, pushing them farther from the demonstration distribution. Despite these pixel-level changes, contact evolution—whether an imprint deepens, slides, or rotates—often remains predictable. We therefore argue for *implicit tactile dynamics modeling*: predicting contact evolution in a dynamics-aware representation space rather than reconstructing future tactile observations.

Latent prediction is a step toward this goal, but not every latent space is equally suited to forecasting tactile evolution. N_0-VTLA (NeoteAI Team et al., 2026b) predicts the net tactile change at a single horizon in the feature space of a frozen DINOv2 (Oquab et al., 2023). However, DINOv2 is trained on static natural images and is not explicitly optimized to capture frame-to-frame tactile evolution; recent tactile representation learning also shows the limitations of single-frame encoders in modeling such dynamics (Feng et al., 2026). In contrast, V-JEPA 2.1 (Mur-Labadia et al., 2026) learns representations by predicting masked spatio-temporal features in videos, encouraging its encoder to capture predictable motion rather than exact pixel details. Building on this objective, we learn a target space from tactile video clips that captures temporal change, while preserving spatial structure to support robust prediction under tactile distribution shifts. Moreover, because a single net change can hide intermediate contact events when a squeeze or slip returns to a similar final state, we predict both future representations and their changes from the present at multiple horizons. This captures contact evolution rather than only its endpoint.

In this paper, we present TacDyn-WAM, a heterogeneous visuo-tactile world action model whose two world-model experts use separate parameters and target spaces. The visual expert predicts future visual latents, while the Implicit Tactile Dynamics Expert (ITDE) predicts tactile evolution in the TacRep space. Joint attention couples their predictions at every layer and allows the Action Expert to use both. The ITDE requires only a single forward pass without iterative denoising. A separate Tactile Understanding Memory provides direct access to the current tactile state, while staged training progressively integrates the tactile modules with the pretrained experts.

On UniVTAC (Chen et al., 2026), TacDyn-WAM achieves state-of-the-art-level performance using only the provided demonstrations. It is competitive with the two strongest models on this benchmark, N_0-VTLA and N_0-TWAM, both of which rely on large-scale visuo-tactile pretraining. Notably, N_0-TWAM follows the video-generation recipe and has roughly twice the parameters of TacDyn-WAM. Ablations show the benefits of both tactile pathways and TacRep's advantage over pixel-reconstruction and static alternatives. On five real-robot tasks, TacDyn-WAM trained on per-task demonstrations alone reaches 71.0% average success, 16 points above the tactile policy FTP-1, and modest-scale pretraining on about 6,000 real visuo-tactile trajectories raises it to 85.0%.

In summary, this work makes the following contributions:

- We propose TacDyn-WAM, a heterogeneous visuo-tactile world action model that jointly predicts future visual latents and tactile evolution in distinct target spaces, while a read-only Tactile Understanding Memory provides the current tactile state.
- We introduce TacRep, a dynamics-aware representation space for implicit tactile dynamics prediction, enabling one-pass multi-horizon prediction without pixel reconstruction or iterative denoising.
- TacDyn-WAM achieves state-of-the-art-level performance on UniVTAC using only the provided demonstrations, and ablations validate the design. On a real robot, it outperforms vision-only and tactile baselines and benefits further from modest-scale tactile pretraining.

2 METHOD

2.1 PROBLEM SETUP AND OVERVIEW

Base Model. We build on the 3B checkpoint of InternVLA-A1 (Cai et al., 2026a), a mixture-of-transformers world action model with three experts coupled through block-wise causal joint attention. The *Understanding Expert*, a Qwen3-VL-2B vision-language model (Bai et al., 2025), encodes the current RGB observations I_t and language instruction ℓ into a prefix block $\mathbf{U}$. The *Visual Generation Expert* receives a visual history block $\mathbf{V}$, encoded by a frozen Cosmos tokenizer (Agarwal et al., 2025), and predicts the future visual latent $\mathbf{z}^v_{t+15}$. The *Action Expert* receives the robot state s_t and a noisy action block $\mathbf{A}$, then denoises an action chunk $\mathbf{a}_t$ of length $H = 50$ through flow

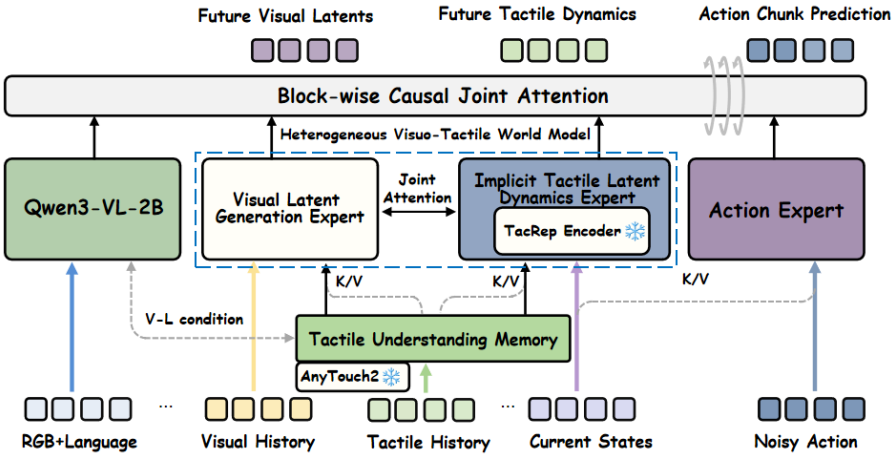


Figure 1: Overview of TacDyn-WAM. The visual and tactile experts form a heterogeneous world model, while the Tactile Understanding Memory provides the current tactile state as read-only keys and values.

matching (Lipman et al., 2022). Let $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ and $\sigma \in [0, 1]$ be the flow-matching time; the noisy chunk is $\mathbf{a}_t^\sigma = \sigma\epsilon + (1-\sigma)\mathbf{a}_t$, and the expert predicts a velocity v_θ toward $\epsilon - \mathbf{a}_t$:

$$\mathcal{L}_{\text{act}} = \mathbb{E}_{\sigma,\epsilon} \left\| v_\theta(\mathbf{a}_t^\sigma, \sigma \mid \mathbf{U}, \mathbf{V}, s_t) - (\epsilon - \mathbf{a}_t) \right\|_2^2 . \tag{1}$$

The visual latent prediction loss $\mathcal{L}_{\text{vis}}$ is the mean squared error to the Cosmos latent of I_{t+15}, and the base model is trained with $\mathcal{L}_{\text{act}} + \lambda_{\text{vis}}\mathcal{L}_{\text{vis}}$.

Problem Setup. We extend the base model with K vision-based tactile sensors, two per gripper. Let $\mathbf{x}_t^k$ denote the reference-subtracted RGB observation (Appendix C) from sensor k at time t, and $\mathbf{x}_{t-3:t}^k$ its four-frame history. Given $\mathcal{C}_t = (\mathbf{U}, \mathbf{V}, \{\mathbf{x}_{t-3:t}^k\}_{k=1}^K, s_t)$, the model predicts

$$\hat{\mathbf{z}}_{t+15}^v = f_{\text{vis}}(\mathcal{C}_t), \qquad \left(\hat{\mathbf{z}}_{t+h}^k, \Delta\hat{\mathbf{z}}_{t+h}^k\right) = f_{\text{tac}}(\mathcal{C}_t), \qquad \hat{\mathbf{a}}_t = f_{\text{act}}(\mathcal{C}_t), \tag{2}$$

where $h \in \mathcal{H} = \{5, 15, 25\}$. The tactile targets are the future representation $\mathbf{z}_{t+h}^k = E(\mathbf{x}_{t+h}^k)$ produced by a frozen dynamics-aware tactile encoder E (Section 2.2) and its change from the present, $\Delta\mathbf{z}_{t+h}^k = \mathbf{z}_{t+h}^k - \mathbf{z}_t^k$. Both world-model predictions use only observed histories and the current state; future observations serve only as training targets.

TacDyn-WAM Overview. TacDyn-WAM adds the *Implicit Tactile Dynamics Expert* (ITDE) as a fourth expert (Figure 1). The ITDE operates on a tactile block $\mathbf{T}$ encoded by the frozen *TacRep Encoder* E and predicts future tactile representations and their changes at multiple horizons. Together with the *Visual Latent Generation Expert* of the base model, it forms a *Heterogeneous Visuo-Tactile World Model*: the two experts use separate parameters and target spaces while interacting through joint attention at every layer.

A separate *Tactile Understanding Memory* compresses the same tactile history into prefix-conditioned tokens and exposes them only as keys and values $\mathbf{M}$, avoiding a fifth expert. The *Action Expert* can therefore combine predicted visual and tactile futures with the current tactile state. Three stage-specific causal masks gradually open the connections between the tactile modules and pretrained experts, as detailed in Section 2.5.

2.2 TacRep: A Dynamics-Aware Tactile Representation Space

Why the Target Space Matters. As argued in Section 1, deployment drift can expose a policy to tactile images outside the demonstration distribution. Their pixels may change substantially even when the underlying physical trend remains broadly consistent. Reconstructing future pixels or VAE latents ties predictions to these unstable pixel patterns and may therefore produce unreliable forecasts under such shifts. We instead let the ITDE predict tactile dynamics in a representation space designed to capture contact evolution while tolerating pixel-level deviations. Following V-JEPA 2.1 (Mur-Labadia et al., 2026), which learns temporal and physical regularities from video, we adapt its encoder to tactile video clips so that the resulting features capture how an imprint

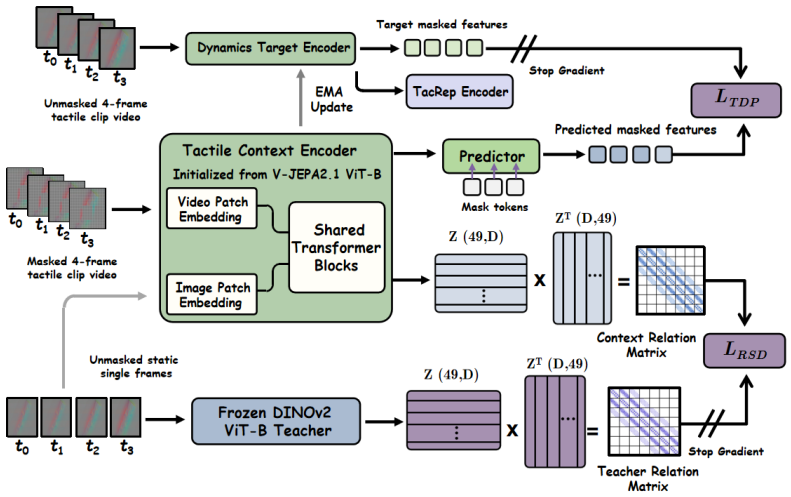


Figure 2: Stage 1: TacRep training. Tactile Dynamics Prediction trains the encoder against an EMA target on masked four-frame clips. Relational Structure Distillation keeps local patch relations close to a frozen DINOv2. Only the EMA encoder is kept.

moves and deforms. A frozen DINOv2 (Oquab et al., 2023) then serves as a structural teacher, regularizing local patch relations to support robust prediction when tactile inputs deviate from the demonstrations.

Tactile Dynamics Prediction. Figure 2 illustrates the training. The *Tactile Context Encoder* E_θ is initialized from the distilled V-JEPA 2.1 ViT-B checkpoint. It takes a four-frame tactile video clip $\mathbf{X} = \mathbf{x}^k_{t-3:t}$ and groups the frames into tubelets of two, so a clip becomes a $2 \times 24 \times 24$ grid of patch tokens. The *Dynamics Target Encoder* $\bar{E}$ shares the same architecture and is updated as an exponential moving average of E_θ. Following V-JEPA 2.1, we mask a set of spatio-temporal blocks $\mathcal{M}$, encode the visible tokens with E_θ, and train a light predictor P_ϕ to recover the target-encoder features of the masked tokens:

$$\mathcal{L}_{\text{TDP}} = \mathbb{E}_{\mathcal{M}} \Big\| P_\phi\big(E_\theta(\mathbf{X} \setminus \mathcal{M}), \mathcal{M}\big) - \text{sg}\left[\bar{E}(\mathbf{X})\right]_{\mathcal{M}} \Big\|_1, \tag{3}$$

where $\text{sg}[\cdot]$ denotes stop-gradient. We use two mask views from V-JEPA 2.1, one with many small blocks and one with few large blocks, and add the same loss on the visible tokens with weight 0.5. Since every masked block spans all four frames, the encoder can recover it only by understanding how the imprint moves across the clip. The resulting features therefore respond to the trend of the contact rather than to the static pixel pattern of the gel.

Relational Structure Distillation. TDP captures temporal evolution but does not explicitly preserve local spatial relations, which may become unstable when tactile inputs deviate from the demonstrations. We therefore use a frozen DINOv2 ViT-B as a *Relational Structure Teacher* to regularize these relations, while temporal dynamics remain learned through TDP. For RSD, the same four frames are processed independently as unmasked static tactile images. The Tactile Context Encoder processes them through its image path with tubelet size one, and the teacher processes them with its own input normalization. Each frame yields a feature grid that we pool to 7×7 and normalize. Let $\mathbf{z}_i$ and $\mathbf{d}_i$ be the features of patch i from E_θ and the teacher, respectively. Since V-JEPA 2.1 and DINOv2 define different latent spaces, directly regressing $\mathbf{z}_i$ to $\mathbf{d}_i$ would create conflicting feature targets, pulling the tactile features toward a static image representation and weakening the dynamics learned by TDP. Instead, we align their local cosine relations, transferring spatial structure without forcing the feature values to match:

$$\mathcal{L}_{\text{RSD}} = \frac{1}{|\mathcal{N}|} \sum_{(i,j)\in\mathcal{N}} \text{SmoothL1}\left(\mathbf{z}_i^\top \mathbf{z}_j - \mathbf{d}_i^\top \mathbf{d}_j\right), \tag{4}$$

where $\mathcal{N}$ contains all patch pairs that lie within two rows and two columns of each other on the 7×7 grid. The Stage 1 objective is $\mathcal{L}_{\text{TDP}} + \lambda_{\text{RSD}}\mathcal{L}_{\text{RSD}}$, with a small λ_{RSD} so that RSD regularizes spatial structure without dominating TDP. After training, the predictor and teacher are discarded,

while the Dynamics Target Encoder $\bar{E}$ is retained as the frozen *TacRep Encoder* E for subsequent ITDE training.

2.3 Implicit Tactile Dynamics Expert

Context Tokens and Future Queries. As shown in Figure 3, the frozen TacRep Encoder converts the four-frame tactile clip from each sensor into two temporal feature maps. Each map is average-pooled to a 4×4 grid and projected to the ITDE hidden dimension, yielding 32 *TacRep Context Tokens* per sensor. For each of the three prediction horizons $h \in \mathcal{H} = \{5, 15, 25\}$ frames, we append 16 learnable *Future Token Queries* arranged on a 4×4 spatial grid. Both context and future tokens receive spatial and modality embeddings, together with a continuous time embedding determined by their frame offsets. The current robot state s_t is injected after every Transformer layer through a zero-initialized conditioning block.

Dynamics Predictor. The 80 tokens per sensor pass through a 28-layer Transformer with hidden size 1024, about 0.4B parameters. It has the same layer geometry as the visual and action experts so that all experts can share attention, but it is initialized from scratch. The outputs of the 32 context tokens and the 48 future tokens together form the tactile block $\mathbf{T}$ that the other experts read.

Future and Delta Representations. The future tokens are supervised through two independent fine-grained readouts, so the 16 compact tokens of each horizon are expanded to a 7×7 prediction outside the 0.4B predictor. The *Future Readout* and the *Delta Readout* each hold 49 learnable queries on a 7×7 grid and cross-attend to the 16 future tokens of one horizon, followed by a linear head to the TacRep dimension. The targets come from the frozen TacRep Encoder applied to single frames. For horizon h, we encode the current frame and the future frame, pool the feature maps to 7×7, and normalize each patch:

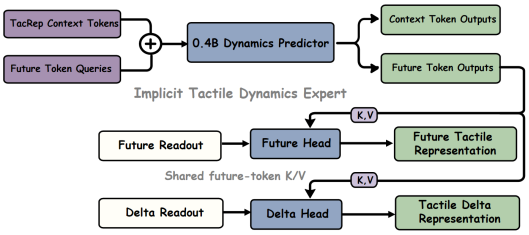


Figure 3: Implicit Tactile Dynamics Expert. A 0.4B dynamics predictor maps context tokens and future queries to the Future and Delta representations at three horizons.

$$\mathbf{z}^k_{t+h} = \mathrm{norm}\left(\mathrm{pool}(E(\mathbf{x}^k_{t+h}))\right), \qquad \Delta\mathbf{z}^k_{t+h} = \mathbf{z}^k_{t+h} - \mathbf{z}^k_t. \tag{5}$$

The Future Readout predicts the *Future Tactile Representation* $\mathbf{z}^k_{t+h}$, with its output normalized in the same way, and the Delta Readout predicts the *Tactile Delta Representation* $\Delta\mathbf{z}^k_{t+h}$. Predicting only the future representation may encourage memorization of typical end states, whereas a delta alone is ambiguous without its starting tactile state. Predicting both requires the ITDE to model how the current tactile state evolves into the future.

Patch-Adaptive Loss Reweighting. Both readouts are optimized with a Smooth L1 loss. Since most patches remain nearly unchanged over a prediction horizon, an unweighted average would be dominated by static regions. We therefore weight the loss of each patch according to the magnitude of its ground-truth change:

$$\mathcal{L}_{\text{tac}} = \frac{1}{2|\mathcal{H}|P} \sum_{h\in\mathcal{H}} \sum_{p} w_{h,p} \Big[\mathrm{SmoothL1}\left(\hat{\mathbf{z}}_{t+h,p} - \mathbf{z}_{t+h,p}\right) + \mathrm{SmoothL1}\left(\Delta\hat{\mathbf{z}}_{t+h,p} - \Delta\mathbf{z}_{t+h,p}\right) \Big],$$
$$w_{h,p} \propto 1 + \alpha \left\|\Delta\mathbf{z}_{t+h,p}\right\|_2. \tag{6}$$

Here, p indexes the $P = 49$ patches of each sensor, and the loss is further averaged over sensors. Within each prediction horizon, the weights are normalized to have a mean of one and capped at a maximum value. This weighting focuses learning on regions undergoing contact expansion, contraction, or sliding, while retaining basic supervision for static patches. All predictions are produced in a single forward pass without iterative denoising. The readouts are used only during training; at deployment, the *Action Expert* directly reads the tactile block $\mathbf{T}$.

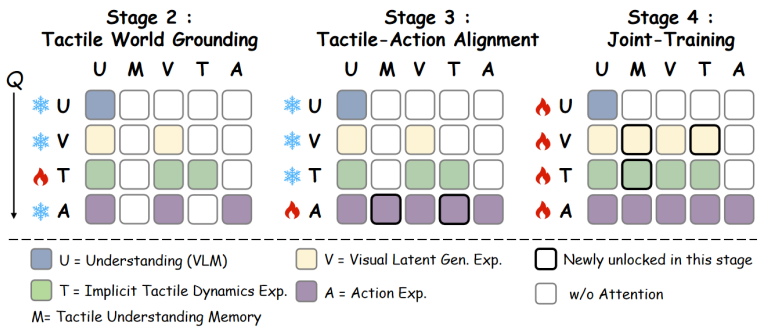


Figure 4: Attention visibility and trainable modules in Stages 2 to 4. Rows are queries and columns are keys; U, M, V, T, and A denote the prefix, memory, visual, tactile, and action blocks. Bold marks connections unlocked in each stage and a flame marks trained modules.

2.4 Tactile Understanding Memory

Compact Tactile Memory. The understanding pathway summarizes the current tactile state: we encode the same four-frame tactile clip with a frozen AnyTouch2 encoder (Feng et al., 2026) pretrained on a large multi-sensor corpus. Since its output contains several hundred tokens per sensor, the *Compact Attentive Pooler* compresses them into 10 tokens: one global token from the class embedding and nine local tokens obtained by pooling the latest temporal feature map into a 3×3 grid. These tokens are refined through cross-attention over the full AnyTouch2 sequence. A two-layer *Compact Tactile Q-Former* then fuses tokens across sensors through self-attention and conditions them on the prefix $\mathbf{U}$ through cross-attention. We use AnyTouch2 only in the understanding pathway: its reconstruction-based pretraining preserves fine-grained tactile information about the current contact, whereas the ITDE operates in the TacRep space designed specifically for predicting future tactile dynamics without pixel reconstruction.

Read-Only Integration. The memory tokens stay outside the residual streams of all experts. Each joint-attention layer maps them into additional keys and values through its own lightweight projections, and only the experts permitted by the causal mask can read them. Because the memory contributes no queries, its tokens are never updated. They occupy fixed position IDs right after the prefix, add only about 1% to the parameter count, and leave the original blocks unaffected when masked out.

2.5 Heterogeneous Joint Attention and Staged Training

Joint Attention. At each layer, the prefix, visual, tactile, and action blocks compute separate queries, keys, and values, which are concatenated and processed by joint attention under a blockwise causal mask (Figure 4). In the final stage, the prefix attends only to itself; the visual and tactile blocks attend to the prefix, memory, and both world model blocks; and the action block attends to all context and causally to itself.

Staged Training. Because the ITDE, tactile memory, and readouts are randomly initialized, training the entire model from the outset could disrupt the pretrained experts. We therefore use four stages and gradually open their attention connections. *Stage 1: TacRep Training* learns the tactile representation as described in Section 2.2 and produces the frozen TacRep Encoder used by all later stages. *Stage 2: Tactile World Grounding* trains the ITDE and its readouts with $\mathcal{L}_{\text{tac}}$, while the other experts remain frozen; the ITDE reads the prefix and visual block but is not yet visible to them. *Stage 3: Tactile–Action Alignment* trains the tactile memory and *Action Expert* with $\mathcal{L}_{\text{act}}$, allowing the action block to read both tactile pathways. *Stage 4: Joint Training* opens the remaining connections and optimizes all trainable modules with

$$\mathcal{L} = \mathcal{L}_{\text{act}} + \lambda_{\text{vis}}\mathcal{L}_{\text{vis}} + \lambda_{\text{tac}}\mathcal{L}_{\text{tac}}. \tag{7}$$

Retaining $\mathcal{L}_{\text{tac}}$ preserves the predictive role of the tactile block and prevents it from becoming an action shortcut. Training details and hyperparameters are provided in Appendix D.

Table 1: Success rates (%) on the eight UniVTAC tasks. Baselines are grouped into vision-only policies, visuo-tactile policies, and policies with large-scale visuo-tactile trajectory pretraining. Gray rows are excluded from comparison, "–" marks results not reported, and the best participating result per column is bolded.

	Method	Insert Hole	Insert Tube	Lift Can	Pull-out Key	Put Bottle	Lift Bottle	Grasp Classify	Insert HDMI	Avg.
Vision-only Policies	$\pi_{0.5}$ (Physical Intelligence et al., 2025)	25	74	6	35	34	**100**	49	8	41.4
	StarVLA-α (Ye et al., 2026b)	52	69	65	51	**88**	32	68	24	56.1
	InternVLA-A1 (Cai et al., 2026a)	70	56	58	67	37	58	88	12	55.8
	Xiaomi-Robotics-0 (Cai et al., 2026b)	**96**	**98**	13	80	12	21	45	**69**	54.3
	GigaWorld-Policy (Ye et al., 2026a)	12	9	0	32	21	38	20	0	16.5
	LingBot-VA (Li et al., 2026)	42	96	0	58	0	0	17	38	31.4
	Fast-WAM (Yuan et al., 2026c)	66	98	0	73	35	21	72	19	48.0
Visuo-tactile Policies	ACT+UniVTAC (Chen et al., 2026)	24	56	29	46	31	71	99	28	48.0
	VITaL (George et al., 2025)	25	34	8	47	32	72	**100**	6	40.5
	RDP (Xue et al., 2025)	23	75	12	18	41	84	–	–	42.2
	TacForcing (Zhou et al., 2026a)	69	79	63	48	43	90	–	–	65.3
	Tactile-WAM (Wu et al., 2026)	20	85	10	20	45	5	55	0	30.0
Visuo-tactile Pretrained Policies	FTP-1 (Yuan et al., 2026a)	64	79	65	48	47	97	99	4	62.9
	N_0-VTLA (NeoteAI Team et al., 2026b)	95	99	88	99	60	99	100	25	83.1
	N_0-TWAM (20% data)	78	96	80	25	46	51	95	52	65.4
	N_0-TWAM (NeoteAI Team et al., 2026a)	99	98	93	79	87	58	94	68	84.5
	TacDyn-WAM (Ours)	91	97	**87**	**97**	72	97	99	12	**81.5**

3 Experiments

3.1 Experimental Setup

Benchmark. UniVTAC (Chen et al., 2026) is a challenging simulation benchmark for contact-rich manipulation, featuring a parallel gripper with a visuo-tactile sensor on each finger. It comprises eight tasks spanning pose reasoning, shape perception, and precise contact interaction (Table 1).

Baselines. Table 1 compares TacDyn-WAM against vision-only and tactile baselines, including three tactile methods with large-scale tactile pretraining; we report their published UniVTAC results and re-evaluate our base model InternVLA-A1 under our own protocol (Appendix F).

Protocol and Implementation. We use the marker-free tactile RGB images from the GelSight Mini sensors on both gripper fingers, together with a head camera and a wrist camera. Following the benchmark, we train one policy per task on the demonstrations it provides for that task and report the success rate over 100 rollouts per task. Training details are given in Appendix D and evaluation details in Appendix H.

3.2 Main Results on UniVTAC

Table 1 reports the results. TacDyn-WAM reaches an average success rate of 81.5%, second only to the two N_0 models, and is within 3 points of the best baseline on Insert Tube, Pull-out Key, Lift Bottle, and Grasp Classify.

Comparison with Vision-Only Models. TacDyn-WAM improves its base model InternVLA-A1 by 25.7 points and exceeds StarVLA-α, the strongest vision-only VLA, by 25.4 points. The vision-only world action models fall further behind, with the best of them, Fast-WAM, at 48.0%, and TacDyn-WAM reaches at least 72% on seven of the eight tasks.

Comparison with Tactile Models. The UniVTAC policy and VITaL, which add a tactile encoder to ACT, reach 48.0% and 40.5%; RDP and TacForcing add execution-time tactile adaptation and reach 42.2% and 65.3%, while Tactile-WAM, with tactile asymmetric attention, reaches 30.0%. FTP-1 is pretrained on about 3,000 hours of tactile data, yet stays 18.6 points below TacDyn-WAM. N_0-VTLA and N_0-TWAM are pretrained on more than 30,000 hours of visuo-tactile trajectories from six embodiments and 450 tasks, and N_0-TWAM has 7.2B parameters. TacDyn-WAM has about 3.6B parameters and is trained on the per-task demonstrations alone, yet it is only 1.6 and 3.0 points behind them, and it beats N_0-TWAM on Pull-out Key, Lift Bottle, and Grasp Classify. When N_0-TWAM is pretrained on 20% of its corpus, its average drops to 65.4%, 16.1 points below ours. The remaining gaps to N_0-TWAM are on Insert Hole and Put Bottle, where its 20% variant also drops

Table 2: Ablation study on UniVTAC (success rate, %). All variants share the same base model, data, and per-stage training hyperparameters. More details are given in Appendix E.

Method	Insert Hole	Insert Tube	Lift Can	Pull-out Key	Put Bottle	Lift Bottle	Grasp Classify	Insert HDMI	Avg.
w/o Tactile World Model	79	72	70	88	52	76	95	11	67.9
w/o Tactile Understanding Memory	84	83	77	82	58	86	91	9	71.3
TacRep → Cosmos VAE	74	69	63	61	47	70	97	15	62.0
TacRep → DINOv2	85	83	80	89	61	84	98	12	74.0
TacDyn-WAM (full)	**91**	**97**	**87**	**97**	**72**	**97**	**99**	**12**	**81.5**

sharply, and on Insert HDMI, whose short and nearly perfect demonstrations contain almost no tactile variation, so success depends on vision and N_0-TWAM likely benefits from its fine-grained video prediction.

3.3 Ablation Study

Table 2 evaluates the contributions of the two tactile pathways through separate ablations and compares TacRep with two frozen alternatives: the base model's Cosmos VAE and a single-frame DINOv2 ViT-B encoder.

Target Space. The Cosmos VAE performs worst at 62.0%, falling 5.9 points below the variant without tactile prediction. The largest drop occurs on Pull-out Key (61.0%), where the tactile observations in failed rollouts differ clearly from those in the demonstrations (Appendix B). Predicting in a pixel-reconstruction space therefore provides unreliable guidance once execution deviates from the demonstrations. DINOv2 recovers the average to 74.0% but remains 7.5 points below the full model, with the largest gaps on Insert Tube, Put Bottle, and Lift Bottle, which involve sustained rotation of the contact pattern. Static features are less tied to exact pixel patterns but lack the temporal sensitivity provided by TacRep.

Two Tactile Pathways. Removing the tactile world model or memory reduces the average success rate by 13.6 and 10.2 points, respectively, confirming that both pathways contribute. Tactile prediction provides the largest gains on Insert Tube, Put Bottle, and Lift Bottle, where anticipating contact rotation is important. The memory instead supplies direct cues about the current tactile state, notably for Pull-out Key and Grasp Classify.

When Tactile Modeling Offers Limited Benefit. On Insert HDMI, all variants remain within three points of the full model, since its demonstrations contain little tactile variation (Section 3.2). Grasp Classify is mainly a perception task, so only removing the Tactile Understanding Memory causes a clear drop (91%).

3.4 Modest-Scale Tactile Pretraining

We further pretrain TacDyn-WAM on a modest amount of real visuo-tactile data as the starting point for the real-robot experiments in Section 3.5.

Data. OmniViTac (Zheng et al., 2026) is a real-world visuo-tactile dataset of 21,879 trajectories over 86 tasks, collected with an xArm-7 and a handheld TacUMI device with the same parallel-jaw gripper. Since our real-robot tasks are mostly related to assembly, we select a subset of 6,000 trajectories: 2,500 from assembly, 2,000 from grasping, 1,000 from wiping, and 500 from adjustment, leaving out cutting and peeling. We also include the 300 demonstrations that we collected for the five real-robot tasks in Section 3.5, 60 per task. Both sources use the same Xense vision-based tactile sensor on each finger, and we directly use its output tactile difference images, which already remove the no-contact background.

Procedure. Pretraining follows the four stages of Section 2.5, with the two data sources entering at different stages. Stages 1 and 2 train the TacRep Encoder and the ITDE without action supervision. We include the real-robot demonstrations in these two stages, with oversampling, so that the tactile representation and dynamics prediction already cover the contacts of the five real tasks before finetuning. Stages 3 and 4 require action supervision and use OmniViTac only. Although its end-effector actions differ from the joint actions of our robot, these stages let the experts learn to work together on real contact data: the Action Expert learns to read the tactile block and the Tactile Understanding

Table 3: Real-world success rates (%) over 20 trials per task. All baselines are fine-tuned per task on the same demonstrations.

Method	Stack Cups	Remove Plug	Insert Plug	Unscrew Cup Lid	Wipe Whiteboard	Avg.
LingBot-VA	40	10	45	15	50	32.0
InternVLA-A1	55	20	45	10	70	40.0
FTP-1	65	45	65	20	80	55.0
TacDyn-WAM	80	60	80	40	95	71.0
TacDyn-WAM (pretrained)	**95**	**85**	**90**	**55**	**100**	**85.0**

Table 4: Inference latency on one A100 GPU with batch size 1. LingBot-VA (naive) recomputes the history at every step; LingBot-VA (optimized) uses the deployment settings of its paper.

Method	Actions per chunk	Latency per chunk (ms)	Latency per action (ms)
LingBot-VA (naive)	40	2,211	55.3
LingBot-VA (optimized)	40	1,338	33.5
TacDyn-WAM (ours)	50	**484**	**9.7**

Memory, and the tactile modules adapt jointly with the pretrained experts. The two action spaces occupy separate slots of a shared action vector, and after pretraining the model is fine-tuned per task with Stage 4 on its 60 real-robot demonstrations. Data statistics and hyperparameters are listed in Appendix D.2.

3.5 Real-World Experiments

Setup. We use a Franka Research 3 arm with a parallel gripper, one Xense vision-based tactile sensor on each finger, and a wrist camera and a third-person camera. We collect 60 teleoperated demonstrations for each of five contact-rich tasks: Stack Cups, Remove Plug, Insert Plug, Unscrew Cup Lid, and Wipe Whiteboard (Appendix G). We compare with two vision-only world action models, LingBot-VA and InternVLA-A1, and with FTP-1, a tactile policy pretrained on about 3,000 hours of tactile data, all fine-tuned per task on the same demonstrations. TacDyn-WAM is evaluated in two variants: trained per task through the four stages as in simulation, or pretrained as in Section 3.4 and then fine-tuned per task with Stage 4. Each policy runs 20 trials per task from randomized initial configurations (Appendix H.2).

Results. Without the OmniViTac pretraining stage, TacDyn-WAM reaches 71.0% (Table 3), 16 points above FTP-1 and 31 points above its base model InternVLA-A1. The gains are largest on Remove Plug and Unscrew Cup Lid, where the vision-only policies often slip during pulling or tip the bottle over when the lid is not turned to the right position. On Stack Cups, vision-only policies tend to keep pushing downward in a wrong direction, whereas TacDyn-WAM lifts the cup and realigns it when the contact does not match the expected evolution. Modest-scale pretraining improves every task and raises the average to 85.0%, with the largest gains on Remove Plug and Stack Cups, while Unscrew Cup Lid remains the hardest task for all methods.

Inference Latency. Table 4 compares TacDyn-WAM with LingBot-VA, which predicts future video frames by iterative denoising, under the same GPU and synchronous setting (Appendix H.2). Even with the deployment optimizations of its paper, LingBot-VA needs 2.8 times the latency of TacDyn-WAM for a shorter chunk, and its latency grows as the episode history accumulates. TacDyn-WAM predicts in compact latent spaces without iterative video denoising, so its cost is lower and constant over an episode.

4 Conclusion and Limitations

We presented TacDyn-WAM, a heterogeneous visuo-tactile world action model that predicts future visual latents and tactile dynamics in separate target spaces and couples them through joint attention. On UniVTAC, TacDyn-WAM reaches state-of-the-art-level performance with per-task demonstrations alone, and modest-scale pretraining further improves real-robot manipulation. One limitation is that TacRep and the ITDE are trained for a single type of visuo-tactile sensor; generalizing them across tactile sensors, including non-image modalities such as force and taxel arrays, is left for future work.

AI USE STATEMENT

We used generative AI tools to aid and polish the writing of this paper. They also assisted with writing part of the implementation code under the authors' direction. The research questions, method design, core implementation, experiments, analyses, and conclusions are the authors' own, and the authors take full responsibility for the content of the paper.

REPRODUCIBILITY STATEMENT

Several measures have been taken to make our work reproducible. The main paper describes the model architecture, training objectives, staged training procedure, and evaluation setup. The appendix provides the input preprocessing, per-stage hyperparameters, ablation and baseline details, and the evaluation protocol. Simulation experiments are conducted on the public UniVTAC benchmark with publicly available pretrained models. For the real-robot experiments, the appendix provides the task definitions, data-collection procedure, and evaluation criteria. Related demo and core code are provided as supplementary material.

REFERENCES

Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical AI. *arXiv preprint arXiv:2501.03575*, 2025.

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

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

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

Jianxin Bi, Kevin Yuchen Ma, Ce Hao, Mike Shou Zheng, and Harold Soh. VLA-Touch: Enhancing vision-language-action model with dual-level tactile feedback. *IEEE Robotics and Automation Letters*, 11(7):8487–8494, 2026b.

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

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*, 2026a.

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*, 2026b.

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.

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. *IEEE Robotics and Automation Letters*, 2026.

Ruoxuan Feng, Jiangyu Hu, Wenke Xia, Tianci Gao, Ao Shen, Yuhao Sun, Bin Fang, and Di Hu. AnyTouch: Learning unified static-dynamic representation across multiple visuo-tactile sensors. In *International Conference on Learning Representations*, 2025.

Ruoxuan Feng, Yuxuan Zhou, Siyu Mei, Dongzhan Zhou, Pengwei Wang, Shaowei Cui, Bin Fang, Guocai Yao, and Di Hu. AnyTouch 2: General optical tactile representation learning for dynamic tactile perception. In *The Fourteenth International Conference on Learning Representations*, 2026.

Abraham George, Selam Gano, Pranav Katragadda, and Amir Barati Farimani. VITaL pretraining: Visuo-tactile pretraining for tactile and non-tactile manipulation policies. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 258–264. IEEE, 2025.

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.

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

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.

Carolina Higuera, Akash Sharma, Chaithanya Krishna Bodduluri, Taosha Fan, Patrick Lancaster, Mrinal Kalakrishnan, Michael Kaess, Byron Boots, Mike Lambeta, Tingfan Wu, et al. Sparsh: Self-supervised touch representations for vision-based tactile sensing. *arXiv preprint arXiv:2410.24090*, 2024.

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.

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

Yuzhe Huang, Pei Lin, Wanlin Li, Daohan Li, Jiajun Li, Jiaming Jiang, Chenxi Xiao, and Ziyuan Jiao. TaF-VLA: Tactile-force alignment in vision-language-action models for force-aware manipulation. *arXiv preprint arXiv:2601.20321*, 2026b.

Lei Jin, Yiding Ma, Xin Zhang, Chen Gao, Wei Wu, and Yong Li. Tacwam: Anchor-guided world action model with mechanics-aware tactile prediction. *arXiv preprint arXiv:2607.28391*, 2026.

Joshua Jones, Oier Mees, Carmelo Sferrazza, Kyle Stachowicz, Pieter Abbeel, and Sergey Levine. Beyond sight: Finetuning generalist robot policies with heterogeneous sensors via language grounding. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5961–5968. IEEE, 2025.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, et al. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Shengbang Liu, Yueru Jia, Yuyang Yan, Jiaming Liu, Xinran Zhang, Qiuxuan Feng, Yandong Guo, Shiji Zhou, Boxin Shi, and Shanghang Zhang. Taco: Tactile world model as a self-corrector forscalable vla post-training. *arXiv preprint arXiv:2607.02840*, 2026.

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.

Lorenzo Mur-Labadia, Matthew Muckley, Amir Bar, Mido Assran, Koustuv Sinha, Mike Rabbat, Yann LeCun, and Nicolas Ballas. V-JEPA 2.1: Unlocking dense features in video self-supervised learning. In *European Conference on Computer Vision*, pp. 671–689. Springer, 2026.

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

NeoteAI Team, Fudan TEAI Team, et al. N_0-VTLA: Scaling vision-tactile-language-action model with latent tactile tokens. *arXiv preprint arXiv:2607.23782*, 2026b.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

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

Minchi Ruan, LiangQing Zhou, Hongtong Li, Zongtao Wang, ZhaoMing Lu, Jianwei Zhang, and Bin Fang. Retac-act: A state-gated vision-tactile fusion transformer for precision assembly. *arXiv preprint arXiv:2603.09565*, 2026.

Carmelo Sferrazza, Younggyo Seo, Hao Liu, Youngwoon Lee, and Pieter Abbeel. The power of the senses: Generalizable manipulation from vision and touch through masked multimodal learning. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9698–9705. IEEE, 2024.

Shilin Shan, Chuhao Zhou, Ruize Wang, Xinyan Chen, Xiangyu Chen, Xinyu Zhou, Boyu Ma, Iris Yuxuan Hu, Jingliang Li, Celeste Yuxuan Hu, et al. Learning physical interaction: A survey of tactile-and force-aware robot learning. *arXiv preprint arXiv:2608.07558*, 2026.

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*, 2026a.

Yufeng Tian, Shuiqi Cheng, Tianming Wei, Tianxing Zhou, Yuanhang Zhang, Zixian Liu, Qianwei Han, Zhecheng Yuan, and Huazhe Xu. Vitas: Visual tactile soft fusion contrastive learning for visuomotor learning. *arXiv preprint arXiv:2602.11643*, 2026b.

Enyi Wang, Wen Fan, and Dandan Zhang. ADM-DP: Adaptive dynamic modality diffusion policy through vision-tactile-graph fusion for multi-agent manipulation. *arXiv preprint arXiv:2602.21622*, 2026a.

Mingxin Wang, Bin Hu, Bin Qian, Kaitao Jiang, Haoning Wu, Feng Yan, Bowen Jing, Ruiyang Hao, Enyi Wang, Kangning Niu, et al. St-wam: Semantic-temporal world action model for robust manipulation under visual distribution shifts. *arXiv preprint arXiv:2607.28993*, 2026b.

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

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.

Zhengtong Xu, Raghava Uppuluri, Xinwei Zhang, Cael Fitch, Philip Glen Crandall, Wan Shou, Dongyi Wang, and Yu She. Unit: Data efficient tactile representation with generalization to unseen objects. *IEEE Robotics and Automation Letters*, 10(6):5481–5488, 2025.

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

Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Hao Li, Hengtao Li, Jie Li, Jindi Lv, Jingyu Liu, et al. GigaWorld-Policy: An efficient action-centered world–action model. *arXiv preprint arXiv:2603.17240*, 2026a.

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*, 2026b.

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*, 2026c.

Jiawen Yu, Hairuo Liu, Qiaojun Yu, Jieji Ren, Ce Hao, Haitong Ding, Guangyu Huang, Guofan Huang, Yan Song, Panpan Cai, et al. Forcevla: Enhancing vla models with a force-aware moe for contact-rich manipulation. *Advances in Neural Information Processing Systems*, 38:93409–93439, 2025.

Chengbo Yuan, Zicheng Zhang, Mingjie Zhou, Wendi Chen, Yi Wang, Zhuoyang Liu, Dantong Niu, Shuo Wang, Hui Zhang, Wenkang Zhang, et al. FTP-1: A generalist foundation tactile policy across tactile sensors for contact-rich manipulation. *arXiv preprint arXiv:2606.13102*, 2026a.

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*, 2026b.

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*, 2026c.

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.

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*, pp. 100333, 2026a.

Kaidi Zhang, Heng Zhang, Zhengtong Xu, Zhiyuan Zhang, Md Rakibul Islam Prince, Xiang Li, Xiaojing Han, Yuhao Zhou, Arash Ajoudani, and Yu She. Tacvla: Contact-aware tactile fusion for robust vision-language-action manipulation. *arXiv preprint arXiv:2603.12665*, 2026b.

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*, 2026c.

Yuhang Zheng, Songen Gu, Weize Li, Yupeng Zheng, Yujie Zang, Shuai Tian, Xiang Li, Ce Hao, Chen Gao, Si Liu, et al. Omnivta: Visuo-tactile world modeling for contact-rich robotic manipulation. *arXiv preprint arXiv:2603.19201*, 2026.

Jianbo Zhou, Boyuan Zhao, Yuzheng Zhang, Yiyang Chen, Wenxin Chen, Qiuyue Li, Xiangyang Gu, Yuhan Cao, Xiao Xia, Yanzhe Hu, et al. Tacforcing: Streaming action generation with execution-time tactile feedback. *arXiv preprint arXiv:2608.25798*, 2026a.

Jianyi Zhou, Feiyang Hong, Yunhao Li, Yicheng Zhao, Yongjue Cen, Zirui Liu, Jiakang Huang, Zirui Chen, Ruiyang Zhang, Weizhuo Zhu, et al. Touchworld: A predictive and reactive tactile foundation model for dexterous manipulation. *arXiv preprint arXiv:2607.07287*, 2026b.

A Related Work

A.1 Vision–Tactile–Language–Action Models

Vision-Language-Action (VLA) models such as the π series transfer large-scale vision-language pretraining to robot control (Black et al., 2024; Physical Intelligence et al., 2025; Cai et al., 2026a). Because vision alone cannot fully resolve local contact geometry, stability, or interaction forces, vision–tactile–language–action (VTLA) models additionally condition policies on tactile observations. Representative approaches include Tactile-VLA, OmniVTLA, and N_0-VTLA, and further work extends this line to tactile-language control, contact-aware fusion, and tactile–force alignment (Huang et al., 2025; Cheng et al., 2026; NeoteAI Team et al., 2026b; Hao et al., 2025; Zhang et al., 2026a;b; Huang et al., 2026b; Jones et al., 2025). Despite their different fusion mechanisms, these methods use tactile signals mainly for perception, where the policy reacts to the current or recent tactile observation. VLA-Touch further introduces tactile feedback at both the VLM and action-refinement levels, but likewise does not explicitly model future tactile evolution (Bi et al., 2026b). In contrast, TacDyn-WAM combines its current-state Tactile Understanding Memory with an explicit predictive tactile dynamics pathway.

A complementary line uses tactile or force signals for reactive action refinement, ranging from slow–fast policies such as Reactive Diffusion Policy and OmniVTA to residual correction (Xue et al., 2025; Zheng et al., 2026; Bi et al., 2026b; Zhou et al., 2026b; He et al., 2026; Wang et al., 2026c; Zhou et al., 2026a). These methods target rapid feedback after contact, whereas TacDyn-WAM models tactile dynamics inside the world model rather than in an execution-time correction stage.

A.2 Tactile Representation Learning

Tactile representations are commonly learned through reconstruction or alignment across sensors and modalities (Shan et al., 2026). Reconstruction and masked-modeling methods preserve local detail, whereas alignment across sensors and modalities improves transfer and semantic grounding; representative examples include Sparsh, M3L, UniT, AnyTouch, and AnyTouch 2 (Higuera et al., 2024; Sferrazza et al., 2024; Xu et al., 2025; Feng et al., 2025; 2026). Subsequent work extends these ideas to visuo-tactile pretraining, language-grounded fusion, and generalist policies across tactile sensors (George et al., 2025; Jones et al., 2025; Tian et al., 2026b; Yuan et al., 2026a). However, reconstruction ties features to exact sensor pixel patterns, and semantic alignment can discard subtle temporal cues such as slip, compression, or contact rotation. ViTacFormer predicts future touch as an auxiliary policy objective (Heng et al., 2025), but explicit predictive representation learning remains less explored. Inspired by the joint-embedding predictive architecture (JEPA) paradigm (Assran et al., 2025; Mur-Labadia et al., 2026), TacRep instead learns a target space from tactile clips that emphasizes predictable contact evolution without reconstructing future pixels. It further regularizes local patch relations with a structural prior, so that the spatial structure of the contact pattern remains stable when tactile inputs deviate from the demonstrations.

A.3 Tactile World Action Models

World action models (WAMs) couple future-state prediction with action generation. Existing WAMs predominantly model visual or semantic dynamics (Yuan et al., 2026c; Ye et al., 2026c; Bi et al., 2026a; Li et al., 2026; Wang et al., 2026b). Recent tactile WAMs can be distinguished by their prediction targets and inference procedures.

Generative tactile prediction. A prominent family reuses video-generation pipelines, such as N_0-TWAM, Dream-Tac, VTAM, and Tactile-WAM: they encode tactile images with the same video VAE as camera frames and generate future tactile observations through iterative denoising (NeoteAI Team et al., 2026a; Lou et al., 2026; Yuan et al., 2026b; Wu et al., 2026; Huang et al., 2026a; Zhou et al., 2026b; Jin et al., 2026). OmniVTA and VT-WAM instead learn a tactile-specific VAE before generative prediction (Zheng et al., 2026; Tian et al., 2026a). Although these designs retain fine spatial detail, their targets remain reconstruction-oriented and inference requires multiple denoising steps. Moreover, pixel similarity does not always track physical similarity: small tactile-image changes may indicate critical slip or rotation, whereas larger pixel changes may be less relevant to control.

Feature and low-dimensional prediction. TacForeSight and ContactWorld avoid image generation by predicting compact tactile features in one forward pass (Zang et al., 2026; Zhang et al., 2026c), and N_0-VTLA predicts the net change of frozen DINOv2 features over a single horizon (NeoteAI Team et al., 2026b). Force-aware WAMs instead predict numeric interaction signals, such as wrist wrench in FAWAM or finger torque in TACO (He et al., 2026; Liu et al., 2026). These targets reduce prediction cost, but their utility depends on whether the representation preserves the spatial and temporal contact structure needed for control. These approaches shift the central design question from how accurately tactile pixels can be reconstructed to which latent dynamics are useful for choosing actions.

TacDyn-WAM follows the one-pass predictive direction, but predicts in a target space learned for tactile dynamics rather than a frozen or reconstruction-oriented one, and keeps this space separate from the visual latent space of the base model.

A.4 Tactile Fusion Mechanisms

Tactile fusion mechanisms can be broadly grouped into attention conditioning, contact-dependent gating, and expert-structured fusion (Shan et al., 2026). Token-based VTLA models typically insert tactile features into a shared sequence or expose them through self- or cross-attention; Tactile-VLA and OmniVTLA are representative examples (Huang et al., 2025; Cheng et al., 2026; Hao et al., 2025; Zhang et al., 2026a; NeoteAI Team et al., 2026b). This design is expressive, but unrestricted fusion can allow a strong vision-language pathway to dominate optimization or let noisy free-space touch interfere with visual features.

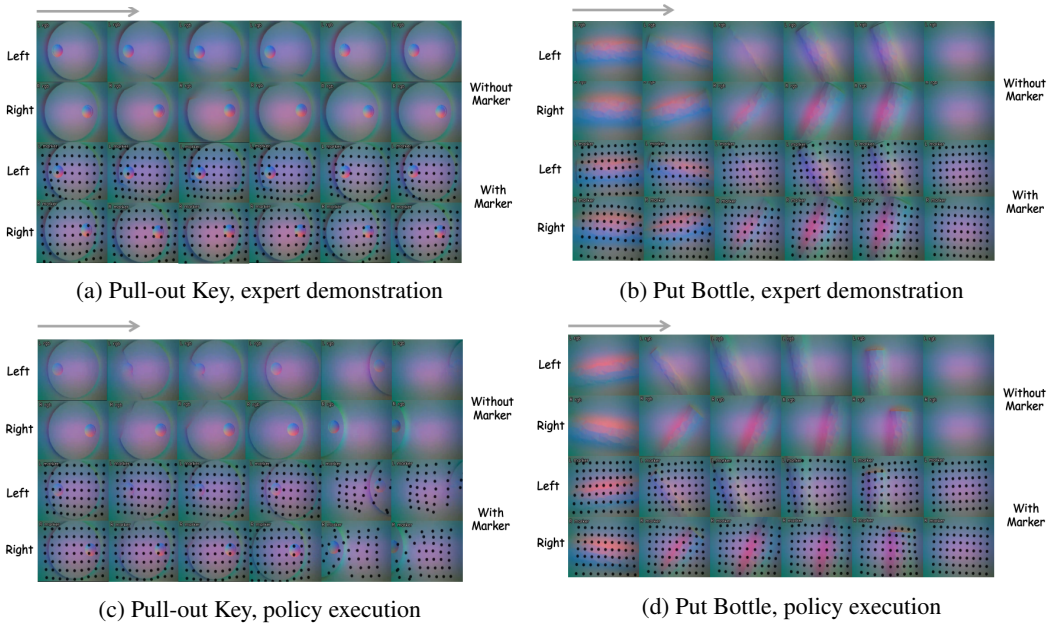
Gated methods make tactile influence dependent on contact state. TacVLA and ReTac-ACT learn gates conditioned on the contact or robot state, while OmniVTA uses a lightweight module to fuse the difference between predicted and observed tactile features (Zhang et al., 2026b; Ruan et al., 2026; Zheng et al., 2026). Similar principles appear as contact-derived attention biases or asymmetric masks in tactile WAMs (Lou et al., 2026; Wu et al., 2026). Expert-structured methods instead preserve modality-specific parameters: ForceVLA uses a force-aware MoE, while N_0-TWAM and VT-WAM separate visual, tactile, and action experts (Yu et al., 2025; NeoteAI Team et al., 2026a; Tian et al., 2026a). Gating and expert separation solve related but different problems: a gate controls the contribution of touch across interaction phases (Wang et al., 2026a), whereas expert separation limits parameter interference between heterogeneous modalities.

TacDyn-WAM likewise specializes parameters by modality, but does not rely on a learned sparse router. Its experts communicate through block-wise joint attention under stage-specific causal masks. In parallel, the Tactile Understanding Memory enters only as read-only keys and values rather than as an additional expert. This design separates current contact understanding, future contact prediction, and action generation, while staged training prevents randomly initialized tactile modules from immediately disrupting the pretrained experts.

B Examples of Tactile Distribution Shift between Demonstrations and Policy Execution

As examples, Figure 5 compares the tactile observations of one expert demonstration with those of one policy execution on Pull-out Key and Put Bottle. Each panel shows six frames from both sensors of the gripper, in both the marker-free and the marker view of the sensor. On Pull-out Key, the robot first rotates the key to the unlocking angle and then pulls it out, so the imprint changes clearly during the rotation phase and mostly jitters sideways before and after it. On Put Bottle, the robot lifts the bottle, rotates it to a suitable angle, and places it on a higher shelf, so the imprint rotates through a sequence of orientations and disappears once the bottle is released. During execution, the pixel pattern departs from the demonstrations: the imprint is off-center, the final rotation angle differs, and the bottle is sometimes released earlier. Yet the overall trend is preserved: the imprint rotates in the same direction as in the demonstration. This is the situation that motivates our target space in Section 2.2: a reconstruction target treats these frames as unseen pixel patterns, while the underlying contact evolution remains predictable. The marker view makes this shift more visible. The marker array deforms with the gel, so the same deviation also displaces every marker near the contact. On Pull-out Key, the later frames of execution show severe marker distortion: the gripper holds the key

tightly and pulls upward while the key remains stuck in the lock, so the gel is stretched and torn instead of following the key. Such deformations never appear in the demonstrations, and they make future tactile images harder to predict and cost extra capacity for a model that reconstructs them.



(a) Pull-out Key, expert demonstration

(b) Put Bottle, expert demonstration

(c) Pull-out Key, policy execution

(d) Put Bottle, policy execution

Figure 5: Examples of tactile distribution shift between expert demonstrations (top) and policy execution (bottom) on Pull-out Key (left) and Put Bottle (right). In each panel, rows show the left and right sensors without and with markers, and columns are six frames over time. The imprint follows a similar trend in both cases, but its position, extent, and timing differ during execution, and the marker view makes the difference more visible.

C Tactile Input Preprocessing

All tactile inputs to the TacRep Encoder and to AnyTouch2 are background-subtracted images rather than raw sensor frames (Figure 6). For each sensor we store one static no-contact frame $\mathbf{b}^k$, captured once with the gripper open, and share it across all tasks. A raw frame $\mathbf{r}_t^k$ is converted to the signed residual

$$\mathbf{x}_t^k = \mathrm{clip}\left(\frac{\mathbf{r}_t^k - \mathbf{b}^k}{255} + \frac{130}{255},\, 0,\, 1\right), \tag{8}$$

so that a pixel equal to the background maps to a constant mid-gray and positive and negative deformations map to brighter and darker values. The subtraction is applied at the native sensor resolution before any resizing, and the same offset is used for both tactile pathways. Raw tactile images contain a fixed gel texture, an illumination gradient, and lens vignetting that are identical in every frame and carry no information about contact. Subtracting the background removes these static patterns, flattens the non-contact area to a uniform value, and leaves only the deformation caused by the current contact, which makes the input closer to what the model should attend to and reduces the patches that the TacRep Encoder must learn to ignore.

D Implementation Details

All stages are trained in bfloat16 with AdamW, a linear warmup followed by cosine decay, and gradient clipping at a norm of 1.0 applied per parameter group. The per-stage training settings of the UniVTAC experiments follow in Appendix D.1.

D.1 UniVTAC Simulation Experiments

Inputs and Actions. At each step the model receives the head and wrist camera frames at $t - 15$ and t, the four most recent tactile RGB frames from both sensors after background subtraction (Ap-

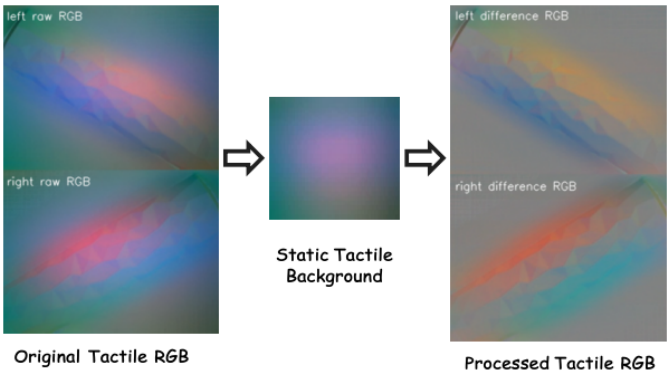


Figure 6: Tactile input preprocessing. Left: raw RGB frames from the two sensors of a gripper. Middle: the static no-contact background frame. Right: the signed residual after subtracting the background and adding a mid-gray offset. Static gel texture and illumination are removed, and only the current deformation remains.

pendix C), together with the robot state and task instruction. Camera frames are 270×480 and tactile RGB frames are 240×320. The model outputs a chunk of 50 actions, with arm joints expressed relative to the current joint angles and an absolute gripper command.

Stage 1. The Tactile Context Encoder is initialized from the distilled V-JEPA 2.1 ViT-B checkpoint with 384×384 inputs. It keeps the two tokenizers of V-JEPA 2.1: the video path embeds a four-frame clip with tubelets of two frames into a $2 \times 24 \times 24$ token grid for TDP, and the image path embeds a single frame with a tubelet of one into a 24×24 grid for RSD; the two paths use separate patch and modality embeddings but share all Transformer blocks. The Relational Structure Teacher is a frozen DINOv2 ViT-B/14 with 392×392 inputs, whose 28×28 patch grid and the student's 24×24 grid are both average-pooled to 7×7 before the relation matrices in Equation 4 are computed; $\lambda_{\text{RSD}} = 0.1$. We use the two multiblock mask views of V-JEPA 2.1, a small view with 8 blocks covering 15% of the spatial area and a large view with 2 blocks covering 70%, both spanning all four frames, with a context loss weight of 0.5. To protect the pretrained representation, the encoder is unfrozen gradually: the predictor and the last four Transformer blocks are trained from the start, blocks 4 to 7 are unfrozen after the first 20% of steps at a lower learning rate, and the first four blocks and the patch embeddings stay frozen throughout. The EMA momentum of the Dynamics Target Encoder increases linearly from 0.999 to 0.9999.

Stage 2. The Future and Delta losses are Smooth L1 with $\beta = 1$, averaged over the feature dimension per patch and combined as in Equation 6. The Patch-Adaptive weights use $\alpha = 2$, are normalized to a mean of one within each sensor and horizon over the batch and the 49 patches, and are clipped at 5 so that no single patch dominates the loss.

Stage 3. The Compact Attentive Pooler, the Compact Tactile Q-Former, and the memory K/V projections use a learning rate of 10^{-4}, and the Action Expert with its state and action projections 5×10^{-5}.

Stage 4. The loss weights are $\lambda_{\text{vis}} = 0.01$ and $\lambda_{\text{tac}} = 500$, where λ_{tac} was set by matching the gradient magnitude of the tactile loss to that of the action loss. Parameters inherited from the base model peak at 5×10^{-5} and decay to 5×10^{-6}; the newly added modules peak at 10^{-4} and decay to 10^{-5}.

Training Schedule. Table 5 lists the optimization settings of the four stages on UniVTAC. The number of Stage 2 and Stage 3 epochs is scaled per task so that tasks with shorter episodes see a comparable number of optimizer steps.

D.2 Modest-Scale Tactile Pretraining

Data. OmniViTac (Zheng et al., 2026) is collected on two embodiments with identical parallel-jaw grippers: an xArm-7 driven by kinesthetic teaching or GELLO teleoperation, and a handheld TacUMI device. We do not distinguish the two. Our subset contains 6,000 trajectories: 2,500 assem-

Table 5: Optimization settings of the four stages on UniVTAC.

	Stage 1	Stage 2	Stage 3	Stage 4
Batch size	64	128	128	64
Peak learning rate	10^{-4} / 10^{-5} / 5×10^{-6}	10^{-4}	10^{-4} / 5×10^{-5}	10^{-4} / 5×10^{-5}
Weight decay	0.04	0.05	0.01	0.01
Warmup	5%	5%	5%	2,000 steps
Length	20 epochs	15 epochs	5 epochs	12,000 steps

bly, 2,000 grasping, 1,000 wiping, and 500 adjustment, all recorded with Xense sensors. The real-robot part consists of the 300 demonstrations of the five tasks in Section 3.5, 60 per task, recorded with the same type of Xense sensor.

Tactile Input Protocol. The Xense sensor outputs tactile difference images with the no-contact background already removed, and both OmniViTac and our real-robot data store this output. We only shift its zero point to the mid-gray offset of $130/255$ used in Equation 8, by subtracting the per-frame median and adding the offset, so that the input has the same form as the simulation residuals. The downstream input protocols are unchanged: AnyTouch2 receives 224×224 inputs with CLIP normalization, and the TacRep Encoder receives 384×384 inputs with ImageNet normalization.

Action Space. The two embodiments occupy non-overlapping slots of a 32-dimensional action vector. Our robot uses joint actions, seven joint deltas and two absolute gripper values in dimensions 0 to 8. OmniViTac uses end-effector actions, the position and 6D rotation relative to the current pose plus an absolute gripper value in dimensions 9 to 18. Each embodiment is normalized with its own statistics, and a validity mask restricts the action loss to the dimensions present in each sample.

Schedule. Table 6 lists the data and training length of each stage. Within OmniViTac, frames are sampled uniformly; in Stages 1 and 2, the real trajectories are oversampled six times, giving them about 23% of the samples. Epochs are counted over unique trajectories. Real-robot fine-tuning starts from the pretrained Stage 4 checkpoint, trains Stage 4 only on the 60 demonstrations of the target task without mixing in pretraining data, and keeps AnyTouch2 and the TacRep Encoder frozen as in simulation.

Optimization. All stages use AdamW with $\beta = (0.9, 0.95)$ and weight decay 0.01, cosine decay, and per-group gradient clipping at a norm of 1.0. Parameters inherited from the base model peak at 5×10^{-5} and the tactile modules at 10^{-4}. The loss weights are $\lambda_{\text{vis}} = 0.01$ and $\lambda_{\text{tac}} = 300$. Pretraining runs on 8 NVIDIA H100 80GB GPUs and real-robot fine-tuning on 4, both in bfloat16 with two gradient accumulation steps and a micro-batch of 16 per GPU in Stage 1 and 8 per GPU in the other stages, which gives the global batch sizes in Table 6. No data augmentation is used.

Table 6: Data and schedule of modest-scale pretraining and real-robot fine-tuning.

	Data	Length	Global batch
Stage 1: TacRep Training	OmniViTac + real	2 epochs	256
Stage 2: Tactile World Grounding	OmniViTac + real	2 epochs	128
Stage 3: Tactile–Action Alignment	OmniViTac	0.5 epochs	128
Stage 4: Joint Training	OmniViTac	3 epochs	128
Real-robot fine-tuning (Stage 4, per task)	60 demonstrations	8,000 steps	64

E Ablation Implementation

All variants in Table 2 share the data, architecture, and per-stage hyperparameters of the full model. The two pathway ablations retrain Stage 4 from the Stage 3 checkpoint of the full model with the attention connections to the removed pathway closed. The two target-space variants replace the frozen TacRep Encoder with the frozen Cosmos tokenizer or DINOv2 ViT-B, keep the same context and target grids, and retrain Stages 2 to 4.

F BASELINES

We compare with the vision-only and visuo-tactile baselines listed in Table 1. Except for InternVLA-A1, all results are taken from the published UniVTAC evaluations of the respective papers.

Vision-only Policies. $\pi_{0.5}$ (Physical Intelligence et al., 2025) is a VLA model built on π_0 that co-trains on heterogeneous data and decodes action chunks with a flow-matching action expert. StarVLA-α (Ye et al., 2026b) is a deliberately minimal VLA design that pairs a strong vision-language backbone with a simple action head. InternVLA-A1 (Cai et al., 2026a) is our base model, a mixture-of-transformers world action model with understanding, visual generation, and action experts (Section 2.1); we re-evaluate it under the same protocol as TacDyn-WAM so that the gain over the base model is measured under identical conditions. Xiaomi-Robotics-0 (Cai et al., 2026b) is an open-source VLA model trained on large-scale cross-embodiment trajectories and vision-language data with asynchronous real-time execution. GigaWorld-Policy (Ye et al., 2026a) is an action-centered world action model that learns pixel-action dynamics with optional video generation. LingBot-VA (Li et al., 2026) is an autoregressive diffusion framework that learns frame prediction and policy execution in a shared latent space with a mixture-of-transformers backbone. Fast-WAM (Yuan et al., 2026c) keeps video co-training during training but skips future prediction at test time. None of these methods uses tactile input.

Visuo-tactile Policies. ACT+UniVTAC (Chen et al., 2026) augments ACT with the UniVTAC tactile encoder, which is pretrained on simulated tactile data with reconstruction and geometric supervision. VITaL (George et al., 2025) pretrains a visuo-tactile encoder and feeds its features to an ACT-style policy. RDP (Xue et al., 2025) is a slow-fast design in which a slow latent diffusion policy predicts action chunks and a fast asymmetric tokenizer performs closed-loop tactile feedback control at high frequency. TacForcing (Zhou et al., 2026a) replaces the action expert of a VLA with a streaming action expert that conditions actions nearing execution on the tactile observations acquired during execution. Tactile-WAM (Wu et al., 2026) is a world action model that predicts future tactile latents with a video VAE and uses asymmetric attention so that video queries do not read tactile keys while action queries do. The last three methods in this group are additionally pretrained on large-scale tactile data. FTP-1 (Yuan et al., 2026a) is a generalist tactile policy that maps image, array, and state-based tactile signals from 21 sensors into unified latent tokens through heterogeneous encoders, pretrained on about 3,000 hours of tactile manipulation data. N_0-VTLA (NeoteAI Team et al., 2026b) builds on $\pi_{0.5}$, encodes tactile difference images with a frozen DINOv2, and conditions the action expert on latent tokens that predict the net tactile change over the action chunk. N_0-TWAM (NeoteAI Team et al., 2026a) is a 7.2B video diffusion world action model with separate video, tactile, and action experts, which predicts future video and tactile frames in a shared VAE latent space by iterative denoising. Both N_0 models are pretrained on more than 30,000 hours of visuo-tactile robot data from six embodiments and 450 tasks; the 20% data variant of N_0-TWAM is pretrained on one fifth of this corpus.

G REAL-WORLD TASKS

Platform. The robot is a Franka Research 3 arm with a parallel gripper. Each finger carries one Xense vision-based tactile sensor, and the scene is observed by a wrist camera and a third-person camera. Demonstrations are collected by teleoperation with a leader arm, 60 per task. Figure 7 shows the five tasks.

Tasks. Each task is specified by a fixed language instruction, and a trial is successful only if the final state meets the criterion below.

- *Stack Cups* ("pick up the paper cup and stack it on top of the other cup"): grasp a paper cup and nest it onto a second cup. The trial succeeds if the cup rests stably inside the other cup. Misaligned cups tend to jam, and a policy that keeps pushing downward fails, whereas a policy that lifts and re-aligns can recover.
- *Remove Plug* ("grasp the plug and pull it out of the socket"): grasp a plug seated in a socket and pull it out. The trial succeeds if the plug leaves the socket while still held. The main failure is slip between the fingers and the plug during pulling.

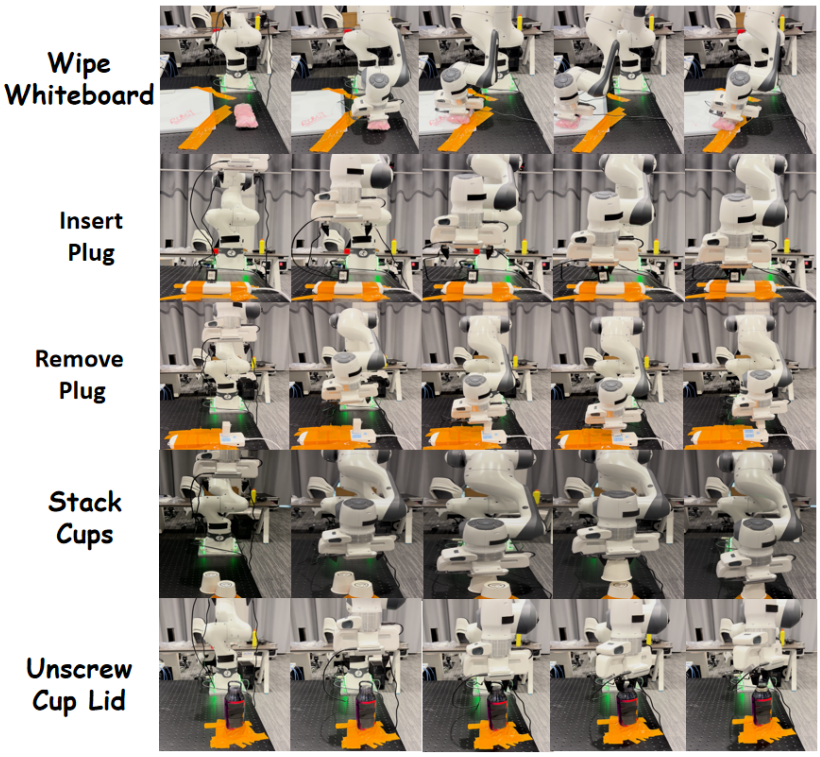


Figure 7: The five real-world tasks. Each row shows one task from left to right over time.

- *Insert Plug* ("grasp the plug and insert it into the socket"): grasp a plug and insert it into a socket. The trial succeeds if the plug is fully inserted. Approaching at a wrong angle either fails to insert the plug or knocks the socket away.
- *Unscrew Cup Lid* ("grasp the cup lid and unscrew it"): grasp the lid of a cup and rotate it until it detaches. The trial succeeds if the lid comes off. Rotating without a proper grip can tip the cup over.
- *Wipe Whiteboard* ("wipe the letters off the whiteboard with the eraser"): grasp an eraser and wipe handwritten letters off a whiteboard. The trial succeeds only if at least 95% of the letters are removed; letters that are not wiped clean or leave a visible residue, as in Figure 8, count as a failure.

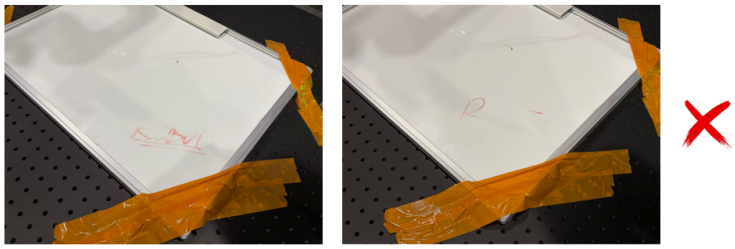


Figure 8: Two failed Wipe Whiteboard trials. Left: the letters are not wiped clean. Right: most of the letters are removed, but part of a letter remains. Both are counted as failures.

H Evaluation Protocol

H.1 Simulation Experiments

Tasks. UniVTAC (Chen et al., 2026) is built on TacEx and Isaac Sim. The robot is a single-arm Franka with a parallel gripper and a GelSight Mini sensor on each finger. Table 7 lists the eight tasks. Expert demonstrations are generated by scripted policies that include randomized failures and

contact-based corrections, and success is judged by physics-based criteria that reject excessive gel penetration and slip.

Table 7: The eight UniVTAC tasks.

Task	Category	Description
Lift Bottle	Pose reasoning	Grasp a bottle standing against a wall and lift it vertically without hitting the wall.
Lift Can	Pose reasoning	Grasp a horizontally placed can of one of three diameters and lift it without slipping.
Put Bottle	Pose reasoning	Grasp a standing bottle and place it into the cavity of a shelf.
Grasp Classify	Shape perception	Touch two visually similar cylinders with different surface textures, then place each on the target pad of its class.
Insert Hole	Contact-rich	Insert a tube into a tilted hole whose orientation must be found by contact.
Insert Tube	Contact-rich	Insert a tube into a narrow hole on a tilted surface with a small clearance.
Insert HDMI	Contact-rich	Align a held HDMI plug with a random initial rotation and insert it into a fixed socket.
Pull-out Key	Contact-rich	Rotate a key in a lock until mechanical resistance is felt, then pull it straight out.

Rollouts. We evaluate one policy per task with 100 rollouts, each starting from a randomized initial configuration drawn by the benchmark. A rollout is scored with the official UniVTAC success criterion of the task. By default the full 50-step chunk is executed before the next inference; on Insert Hole and Lift Can we execute the first 32 steps of each chunk, which we found to give more stable contact. The same protocol is used for our re-evaluation of InternVLA-A1 and for all ablation variants.

H.2 Real-World Experiments

Each policy is evaluated with 20 trials per task. Before each trial, the objects are placed at a random position and orientation within the workspace covered by the demonstrations, and success is judged by the criteria in Appendix G. All policies are trained per task on the same 60 demonstrations. The three baselines are fine-tuned from their released checkpoints. TacDyn-WAM without the OmniVi-Tac pretraining stage is trained through the four stages with the simulation settings, including Stage 1 on the tactile clips of the same demonstrations, and TacDyn-WAM (pretrained) is fine-tuned with Stage 4 from the pretrained checkpoint (Appendix D.2). Both variants predict 50-step action chunks and execute the first 32 steps of each chunk before the next inference.

Latency Measurement. The latencies in Table 4 are measured on the same NVIDIA A100 GPU with batch size 1 and synchronous inference, as the median of 30 runs after warmup, and cover only the GPU computation from the input tensors to the action chunk. LingBot-VA is run in its two-frame chunk setting, which outputs 40 actions per chunk and is closest to our 50-step chunk, with a history of four chunks. Its naive setting runs the full video denoising and recomputes the history at every chunk, while its optimized setting follows the deployment configuration of its paper with partial video denoising and a reused key-value cache. TacDyn-WAM computes the prefix, the memory, and the visual and tactile blocks once per chunk and reuses them across the 10 action denoising steps.