TCR-ARM: TACTILE CONTEXT REASONING AND ACTION REFINEMENT FOR CONTACT-RICH ROBOTIC MANIPULATION

Anonymous authors
Paper under double-blind review

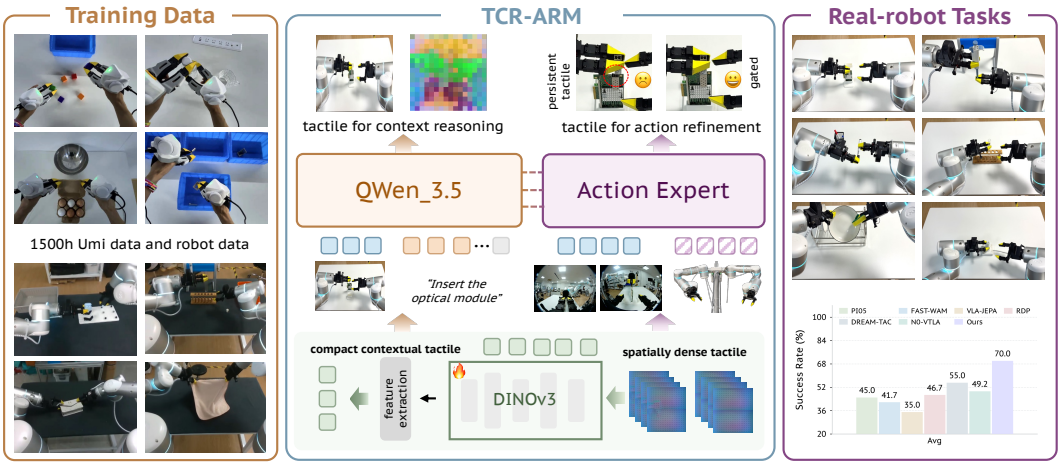

Figure 1: **Overview.** Left: 1,500 hours of UMI and robot data support tactile pretraining. Center: a shared encoder provides compact tactile context for persistent reasoning in the VLM and spatially dense tactile features for gated updates to wrist visual features, supporting selective local action refinement. Unregulated dense refinement can introduce unnecessary action corrections, motivating adaptive regulation of these updates. Right: evaluation across six real-world contact-rich manipulation tasks; TCR-ARM achieves 70.0% average success, compared with 55.0% for the strongest evaluated baseline.

ABSTRACT

Contact-rich robotic manipulation depends on touch to resolve contact states that vision alone may fail to reveal. Yet tactile information should not influence control uniformly. Persistent tactile context can support reasoning about contact state and interaction progress, whereas tactile-driven local action refinement should be applied selectively. We present TCR-ARM, a vision-language-action policy that realizes this distinction through two tactile interfaces derived from a shared encoder. Compact tactile representations support persistent tactile context reasoning in the vision-language model, while spatially dense tactile features adaptively update wrist visual tokens through a learned gate to support local action refinement. A learnable gate alone, however, does not ensure selective use of dense refinement. We therefore introduce route-aware tactile pretraining (RAT), which samples no-tactile, context-only, and full-tactile routes to train tactile context reasoning with and without tactile-driven action refinement. After pretraining on 1,500 hours of tactile manipulation data and post-training with 100 demonstrations per task, TCR-ARM achieves 70.0% average success across six real-world contact-rich robotic manipulation tasks, outperforming the strongest evaluated baseline by 15.0 percentage points. RAT further improves average success by 10.8 percentage points over full-tactile-only training.

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

1 INTRODUCTION

Vision-language-action (VLA) models combine pretrained vision-language knowledge with action learning for instruction-conditioned manipulation (Zitkovich et al., 2023; Black et al., 2025b;a; Kim et al., 2025). Yet recognizing an object and approaching its target pose do not fully determine the motion required after contact. Visually similar insertion configurations can correspond to smooth engagement or a blocked edge, and grasped objects can slip before their displacement becomes visible. Tactile sensing observes the contact interface, supplying information that external images may miss (Yuan et al., 2017). Manipulation policies thus need tactile feedback to resolve ambiguous contact states and guide contact-rich manipulation.

Existing policies use touch for contextual understanding or action generation. Semantic or contact-gated tokens supply contact context (Cheng et al., 2026; Zhang et al., 2026a), while visual feature modulation (Morissette et al., 2026) and dedicated tactile experts or control branches (Niu et al., 2026; Bi et al., 2026) influence action generation. Prior work also combines both interfaces (Bi et al., 2026; Ma et al., 2026) and learns predictive visual or tactile dynamics (Lou et al., 2026; NeoteAI Team & Fudan TEAI Team, 2026a). Adaptive injection and residual correction further regulate tactile influence (Li et al., 2026; Zhang et al., 2026b; Wang et al., 2026). These advances establish complementary interfaces, but access to both does not ensure that each contributes appropriately as contact conditions change.

During insertion, tactile feedback distinguishes smooth engagement from a blocked edge and guides local correction. After release, the absence of contact still indicates interaction progress, while weak or irrelevant tactile feedback can induce unnecessary corrections that disrupt withdrawal. Thus, touch remains useful for understanding contact state and interaction progress even when local action refinement is unnecessary. This motivates persistent tactile context reasoning with selectively applied tactile-driven local action refinement.

TCR-ARM implements this distinction with two tactile interfaces at different spatial granularities derived from a shared encoder (Figure 1). A *compact contextual tactile representation* supports persistent tactile context reasoning in the vision-language model (VLM) and conditions auxiliary future visual latent prediction. A *spatially dense tactile representation* retains local contact structure for action refinement. Wrist visual tokens query this representation, while a wrist-visual-conditioned gate weights the resulting updates before they condition the action expert. The tactile contribution to action refinement can therefore vary with wrist visual context, while compact tactile context remains available throughout full-tactile control.

A learnable gate can regulate tactile updates, but does not by itself ensure selective use of tactile-driven action refinement. When training always uses the full-tactile route, the model is not explicitly trained to perform tactile context reasoning independently of action refinement. We therefore introduce *route-aware tactile pretraining* (RAT), which uniformly samples no-tactile, context-only, and full-tactile routes. Context-only examples train tactile context reasoning without tactile-driven action refinement, while full-tactile examples train both together. Both interfaces remain enabled during post-training and deployment. Our contributions are as follows:

- We introduce TCR-ARM, a VLA architecture for persistent tactile context reasoning and selective local action refinement. A shared encoder supplies compact contextual and spatially dense tactile representations, while a wrist-visual-conditioned gate regulates the contribution of the spatially dense representation to action refinement.
- We propose route-aware tactile pretraining, which uniformly samples no-tactile, context-only, and full-tactile routes to train tactile context reasoning with and without tactile-driven action refinement. RAT improves average success by 10.8 percentage points over full-tactile-only training.
- Across six real-world contact-rich robotic manipulation tasks, TCR-ARM achieves 70.0% average success, exceeding the strongest evaluated baseline by 15.0 percentage points.

2 RELATED WORK

Generalist policies and action learning. RT-2 and OpenVLA transfer visual-language knowledge to robot control (Zitkovich et al., 2023; Kim et al., 2025); π_0 and $\pi_{0.5}$ extend this direction with

continuous actions and heterogeneous training data (Black et al., 2025b;a). Action chunking and diffusion model action sequences (Zhao et al., 2023; Chi et al., 2023), while flow matching supports continuous generation (Lipman et al., 2023) and UMI supports portable demonstration collection (Chi et al., 2024). TCR-ARM builds on these capabilities to incorporate contact information into contextual interpretation and local action generation.

Tactile representations and fusion. Self-supervised learning uses joint visual and haptic observations, human-collected visual-tactile pairs, and large tactile datasets to learn contact representations (Lee et al., 2019; Yang et al., 2022; Higuera et al., 2025). UniTouch and AnyTouch further study cross-modal and cross-sensor alignment (Yang et al., 2024; Feng et al., 2025). Within VLAs, OmniVTLA and TacVLA provide semantic-aligned or contact-aware tactile fusion, while TacFiLM modulates visual features with touch (Cheng et al., 2026; Zhang et al., 2026a; Morissette et al., 2026). Our shared encoder produces compact context and dense local features, assigning these representations to complementary policy interfaces.

Tactile feedback and adaptive control. Tactile-VLA combines tactile reasoning with hybrid position-force control (Huang et al., 2025). Hierarchical feedback and tactile experts support reactive execution (Bi et al., 2026; Xue et al., 2025; Niu et al., 2026), and TacCoRL uses contact-gated tactile tokens in both shared context and action conditioning (Ma et al., 2026). AT-VLA adaptively injects tactile feedback into the action expert (Li et al., 2026); ResTacVLA gates residual tactile representations using visual uncertainty (Zhang et al., 2026b); ViTaR scales bounded action corrections above a frozen VLA (Wang et al., 2026). TCR-ARM retains compact tactile context while gating dense tactile updates to wrist visual tokens that condition action generation. RAT trains the contextual contribution independently of dedicated refinement as well as their joint use.

Predictive learning and tactile route training. V-JEPA 2.1 learns dense video representations through self-supervised prediction (Mur-Labadia et al., 2026). Fast-WAM jointly trains video and action prediction, while VLA-JEPA uses future latent supervision (Yuan et al., 2026; Sun et al., 2026). Dream-Tac models future visual and tactile observations with actions (Lou et al., 2026). N_0-TWAM learns tactile dynamics and applies whole-modality dropout during post-training; N_0-VTLA supplies predictive tactile tokens to the action expert outside the VLM prefix (NeoteAI Team & Fudan TEAI Team, 2026a;b). Modality dropout also has broader precedents (Neverova et al., 2016). TCR-ARM uses tactile-conditioned future visual prediction as an auxiliary objective. RAT samples no-tactile, context-only, and full-tactile conditions to train complementary contributions from both interfaces, with the context-only condition preventing dense refinement from substituting for the contextual contribution during those examples.

3 METHOD

TCR-ARM uses a shared encoder to produce a compact contextual tactile representation for persistent tactile context reasoning and a spatially dense tactile representation for selective local action refinement (Figure 2).

3.1 ARCHITECTURE OVERVIEW

Policy backbone. At control time t, the observation is $o_t = (\ell, I_t^h, I_t^w, s_t, T_t)$: a language instruction, a head-camera image, bilateral wrist-camera images, robot state, and bilateral tactile histories. The policy generates an action chunk $A_t \in \mathbb{R}^{H \times d_a}$ with $H = 40$ and $d_a = 20$. Each arm uses a 10-dimensional end-effector action: 3 position coordinates, a 6D rotation representation, and 1 gripper value. A Mixture-of-Transformers backbone combines a QWen3.5-2B VLM with a continuous action expert. Head visual, language, state, compact tactile, and future task tokens form the VLM prefix; wrist visual tokens enter the action expert alongside noisy action tokens. Shared attention lets the action expert read the prefix, while the prefix cannot read noisy actions.

Shared tactile encoder. A shared encoder initialized from DINOv3 ViT-B/16 (Siméoni et al., 2026) is optimized jointly with the policy. Each side $r \in \{L, R\}$ supplies the current tactile image and three preceding frames. Let F_k^r denote patch features at relative time $k \in \{-3, -2, -1, 0\}$. We

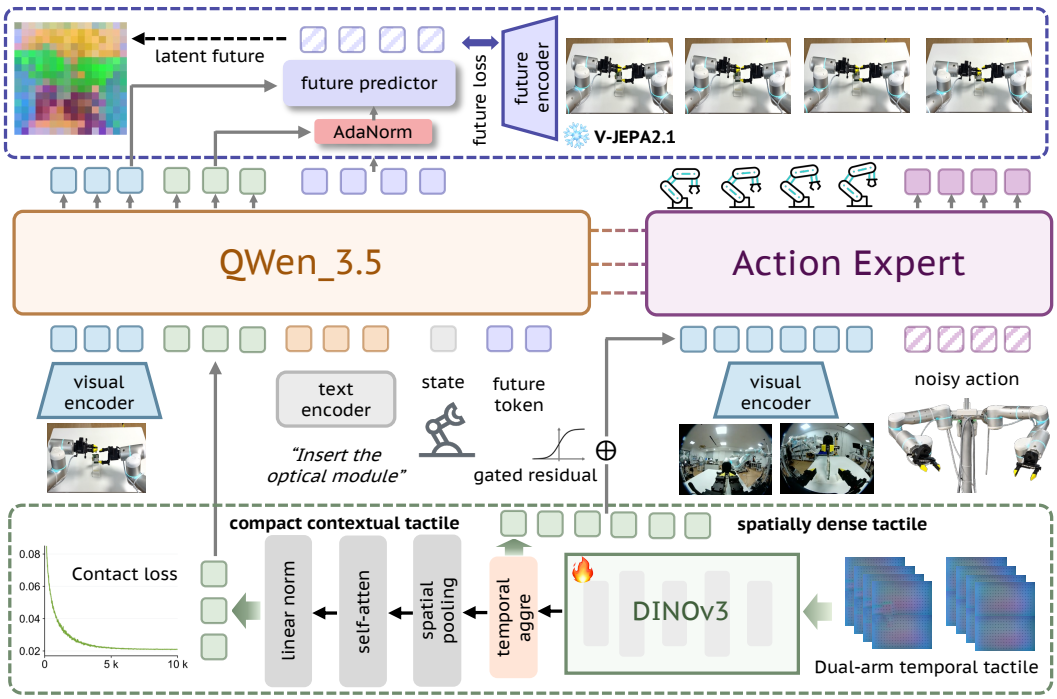

Figure 2: TCR-ARM Architecture. Compact tactile representations provide VLM context and condition future prediction, while spatially dense representations support gated local action refinement. RAT controls route availability during pretraining. Auxiliary modules are inactive at deployment.

add relative-time embeddings, $\widetilde{F}_k^r = F_k^r + e_k$, and concatenate the four frames into $\widetilde{F}_{\text{hist}}^r$. Current-frame queries attend over this history:

$$D^r = \text{Attn}\left(Q_t(\widetilde{F}_0^r), K_t(\widetilde{F}_{\text{hist}}^r), V_t(\widetilde{F}_{\text{hist}}^r)\right), \qquad C^r = \text{Transformer}^{(2)}\left(\text{Pool}_{2\times2}(D^r)\right). \quad (1)$$

Invalid frames and patches are masked. D^r preserves the current query grid, while spatial pooling and two Transformer blocks produce compact tokens C^r. With 128×128 tactile images, bilateral concatenation gives 128 spatially dense tokens D and 32 compact tokens C, projected to their receiving modules' dimensions.

3.2 Tactile Context Reasoning and Action Refinement

Compact contextual tactile representation. Tactile context reasoning integrates touch with vision, language, and robot state to infer contact state and interaction progress. We summarize tactile information through spatial pooling and incorporate the resulting compact representation into the VLM prefix:

$$P = [Z^h; C; E_\ell(\ell); E_s(s_t); Q_f], \quad (2)$$

where Z^h denotes head visual tokens, E_ℓ and E_s embed language and state, and Q_f contains future task tokens used for auxiliary supervision (Section 3.4). Through causal attention, the prefix integrates tactile context with language and robot state for future prediction and action generation. During full-tactile control, this interface remains available independently of the refinement gate.

Spatially dense tactile representation. We retain the spatially dense tactile representation for selective local action refinement. Wrist visual tokens W retrieve tactile content through cross-attention, weighted by a wrist-visual-conditioned gate. With normalized features $\overline{W}$ and $\overline{D}$, the update is

$$R(W, D) = W_o\left[\text{Attn}\left(Q(\overline{W}), K(\overline{D}), V(\overline{D})\right) \odot \sigma\left(G(\overline{W})\right)\right], \quad (3)$$

$$\widetilde{W} = W + m_r R(W, D), \qquad m_r \in \{0, 1\}. \quad (4)$$

The sigmoid gate weights each token and channel before output projection, and the updated wrist tokens condition the action expert. We zero-initialize W_o, mask invalid tactile tokens, and bypass refinement when none are valid. The binary m_r controls route availability during training; the learned gate regulates updates on an enabled route using wrist appearance rather than explicit contact labels.

3.3 Route-Aware Tactile Pretraining

A learnable gate alone does not ensure selective action refinement. Full-tactile-only training does not explicitly train tactile context reasoning without action refinement. RAT addresses this by sampling no-tactile, context-only, and full-tactile routes. Let m_c and m_r control the context and refinement routes:

$$m = (m_c, m_r) \sim \mathrm{Uniform}(\mathcal{S}), \qquad \mathcal{S} = \{(0,0), (1,0), (1,1)\}. \tag{5}$$

The no-tactile state masks tactile prefix slots, zeros tactile conditioning for future prediction, and disables refinement. The context-only state retains tactile context with $m_r = 0$, while the full-tactile state enables both routes. Excluding $(0,1)$ keeps context available whenever action refinement is trained, while $(1,0)$ explicitly trains the contextual contribution without refinement. Sensor-validity masks remain active in every state. Post-training and deployment use $(1,1)$, with the learned gate regulating action refinement.

3.4 Training and Deployment

Auxiliary supervision. Future visual latent prediction provides an auxiliary learning signal for the compact contextual tactile representation. A frozen V-JEPA 2.1 encoder (Mur-Labadia et al., 2026) maps head-camera frames at $t+10$, $t+20$, $t+30$, and $t+40$ to a target Y^*, aligning the endpoint with the 40-step action chunk. Sixteen future task tokens Q_f, obtained by grouped pooling of learnable predictor queries, enter the VLM. Attention pooling over valid contextual tactile hidden states gives $c = \mathrm{AttnPool}(H_C)$. Following feature-wise affine conditioning (Perez et al., 2018), two learned mappings modulate the predictor's output queries q:

$$\widetilde{q} = (1 + \gamma(c)) \odot \mathrm{LN}(q) + \beta(c). \tag{6}$$

The modulated queries attend to head visual and future task hidden states to predict $\widehat{Y}$. Head visual hidden states are detached at the prediction-head input, while tactile and future task hidden states retain gradients.

Binary contact-state supervision further guides the shared encoder. A contact head reads globally pooled C^r and predicts p_c^r for label $y_c^r \in \{0,1\}$. With valid-label mask M_c^r and $N_c = \max(1, M_c^L + M_c^R)$, its loss is

$$\mathcal{L}_c = -\frac{1}{N_c} \sum_{r \in \{L,R\}} M_c^r \left[y_c^r \log p_c^r + (1 - y_c^r) \log(1 - p_c^r) \right]. \tag{7}$$

It averages over valid labeled sides and is zero when neither side has a valid label. This supervision remains active in all RAT states whenever valid labels exist.

Joint learning objective. Given a demonstrated action chunk A^*, Gaussian noise ϵ, and noise level $\tau \in [0,1]$, conditional flow matching (Lipman et al., 2023; Black et al., 2025b) uses

$$A_\tau = (1-\tau)A^* + \tau\epsilon, \quad u^* = \epsilon - A^*, \quad \mathcal{L}_a = \mathrm{MSE}_{M_a}\left(v_\theta(A_\tau, o_t, \tau; m), u^*\right). \tag{8}$$

Writing MSE_M for the mean squared error over valid elements, the joint pretraining objective is

$$\mathcal{L}_{\mathrm{pre}} = \mathbb{E}_{\mathcal{D}_{\mathrm{pre}}, m, \tau, \epsilon}\left[\mathcal{L}_a + \lambda_f \, \mathrm{MSE}_{M_f}(\widehat{Y}, \mathrm{sg}(Y^*)) + \lambda_c \mathcal{L}_c \right], \tag{9}$$

where M_a and M_f mask valid actions and future targets.

Training protocol and deployment. Pretraining uses 1,500 hours of tactile data across 128 tasks: 1,000 hours of UMI data and 500 hours of robot data, spanning organization, insertion and assembly, and fine manipulation. Each target task uses 100 tactile demonstrations for post-training with full

Table 1: Task success rates (%) across six real-world contact-rich manipulation tasks; 20 trials per task. Avg. equally weights the six tasks.

Method	Wipe Plate	Attach Pipette Tip	Insert Optical Module	Cap Pen	Insert Test Tube	Screw Light Bulb	Avg.
$\pi_{0.5}$	30	20	30	55	75	60	45.0
Fast-WAM	20	30	20	50	75	55	41.7
VLA-JEPA	10	15	20	45	70	50	35.0
RDP	20	20	25	70	80	65	46.7
Dream-Tac	35	25	40	80	90	60	55.0
N_0-VTLA	25	20	35	65	85	65	49.2
TCR-ARM	**60**	**40**	**60**	**90**	**100**	**70**	**70.0**

tactile input and the same joint objectives. Action-prefix conditioning prepares the policy for real-time chunking (RTC), with action supervision restricted to the generated suffix (Black et al., 2025c). RTC overlaps execution with asynchronous generation of the next chunk; each request re-encodes the VLM prefix and reuses its attention keys and values during five Euler steps from $\tau = 1$ to 0. At deployment, tactile context and gated action refinement remain active, while the contact head, future predictor, and V-JEPA teacher are inactive.

4 Experiments

4.1 Evaluation Setup

Tasks and metrics. We select six contact-rich real-world manipulation tasks on Flexiv Rizon4 robot arms: Wipe Plate, Attach Pipette Tip, Insert Optical Module, Cap Pen, Insert Test Tube, and Screw Light Bulb. These tasks cover sustained surface contact, attachment, insertion, and threaded assembly. Each method is evaluated over 20 trials per task and condition. Unless stated otherwise, success rates measure full-task completion, and the reported mean equally weights the six tasks.

Models and adaptation. TCR-ARM uses the pretraining and post-training procedure in Section 3.4, including 100 tactile demonstrations per target task and full-tactile deployment. For a fair comparison, each baseline is initialized from its officially released pretrained weights, post-trained using the same 100 demonstrations per target task, and evaluated under the same protocol. We compare against $\pi_{0.5}$ (Black et al., 2025a), Fast-WAM (Yuan et al., 2026), VLA-JEPA (Sun et al., 2026), RDP (Xue et al., 2025), Dream-Tac (Lou et al., 2026), and N_0-VTLA (NeoteAI Team & Fudan TEAI Team, 2026b). These baselines span generalist, predictive, and tactile-informed policies.

4.2 Main Results

Table 1 shows that TCR-ARM achieves 70.0% average success, exceeding Dream-Tac, the strongest evaluated baseline, by 15.0 percentage points. It has the highest success rate on every task, with gains of 5-25 points over each task's strongest baseline. Plate wiping and optical-module insertion show the largest gains: 60% success on each, exceeding their strongest baselines by 25 and 20 points. These results show consistent gains for the complete policy across different contact interactions; the following ablations examine the roles of its tactile interfaces and training strategy.

Figure 3 complements these aggregate results with representative executions of all six tasks. The enlarged views expose local contact configurations within the wider robot observations, while tactile insets show the available contact feedback at selected stages. Across insertion, attachment, wiping, and threaded assembly, the sequences cover both localized engagement and sustained contact during motion. These examples illustrate the contact interactions addressed by the complete policy; the following ablations evaluate how contextual touch and dedicated refinement contribute to its performance.

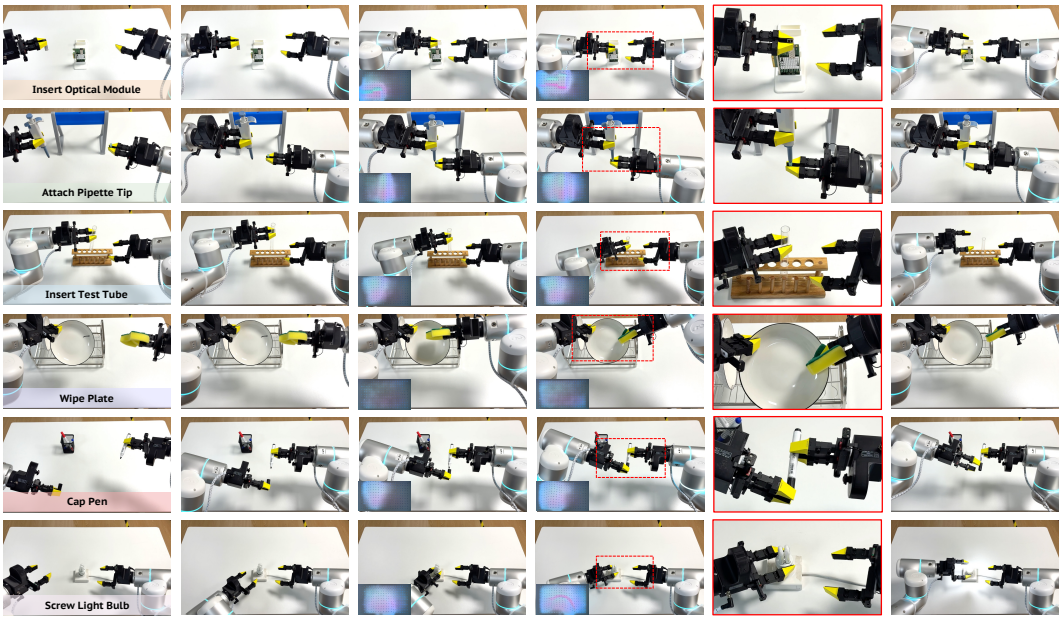

Figure 3: Representative executions across six contact-rich tasks, showing robot observations and tactile insets at selected interaction stages. Red boxes identify contact regions enlarged in the fifth column, highlighting the local configurations involved in insertion, attachment, wiping, and threaded assembly.

Table 2: Ablation of tactile routes. WP: Wipe Plate; PT: Attach Pipette Tip; OM: Insert Optical Module; CP: Cap Pen; TT: Insert Test Tube; LB: Screw Light Bulb. Both-route variants use gated dense refinement.

Tactile configuration	WP	PT	OM	CP	TT	LB	Avg.
No tactile input	30	20	25	50	75	50	41.7
Context route only	30	25	20	60	85	60	46.7
Refinement route only (ungated)	25	15	25	40	75	50	38.3
Refinement route only (gated)	35	25	30	55	80	60	47.5
Both routes (w/o RAT)	50	30	45	70	95	65	59.2
Both routes (with RAT)	**60**	**40**	**60**	**90**	**100**	**70**	**70.0**

4.3 Contributions of Tactile Context and Refinement

Table 2 disentangles the contributions of the context route, refinement route, gating, and route-aware training. The context route alone improves average success from 41.7% to 46.7%, indicating that compact tactile representations provide useful persistent context. In contrast, the ungated refinement route decreases performance to 38.3%, showing that directly injecting dense tactile features does not necessarily improve control. Adding the wrist-visual-conditioned gate raises the refinement-only result to 47.5%, highlighting the importance of selectively regulating local tactile updates. Combining both routes further improves performance to 59.2%, supporting their complementary roles. Finally, RAT increases the average success rate to 70.0%, demonstrating the benefit of coordinated training across the two tactile routes.

4.4 Route-Aware Tactile Pretraining

Table 3 compares RAT with four alternative pretraining strategies. *Full-tactile-only* keeps $(m_c, m_r) = (1, 1)$ throughout. *Whole-modality dropout* samples $(0, 0)$ and $(1, 1)$ with equal probability. *Refinement-preserving sampling* uniformly samples $\{(0, 0), (0, 1), (1, 1)\}$, and *independent route dropout* samples the two routes independently. RAT uniformly samples $\{(0, 0), (1, 0), (1, 1)\}$. Both tactile routes are enabled during post-training and evaluation for all strategies.

Table 3: Ablation of pretraining route selection. Success rates (%) measure full-task completion. Task abbreviations follow Table 2.

Pretraining strategy	WP	PT	OM	CP	TT	LB	Avg.
Full-tactile-only training	50	30	45	70	95	65	59.2
Whole-modality dropout	50	35	50	75	**100**	**70**	63.3
Refinement-preserving sampling	45	25	55	70	**100**	65	60.0
Independent route dropout	55	35	55	80	95	**70**	65.0
Route-aware tactile pretraining	**60**	**40**	**60**	**90**	**100**	**70**	**70.0**

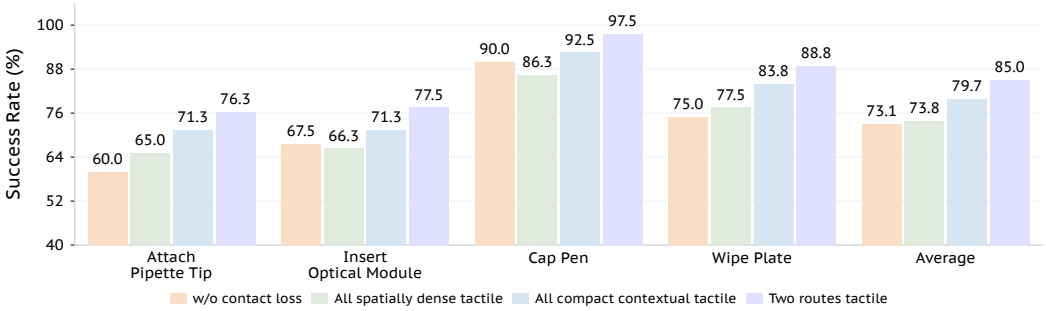

Figure 4: Spatial-granularity ablation across four tasks. Each task bar averages success over its four substages; the final group reports the average across tasks.

RAT reaches 70.0% average success, exceeding always-on training, whole-modality dropout, refinement-preserving sampling, and independent route dropout by 10.8, 6.7, 10.0, and 5.0 percentage points, respectively. RAT trains the contextual contribution without dense refinement in $(1, 0)$ and their joint use in $(1, 1)$. Its advantage over generic dropout and refinement-preserving sampling supports training context independently while retaining it whenever refinement is enabled.

4.5 Spatial Granularity of Tactile Representations

To examine tactile spatial granularity, we divide four tasks into four substages and conduct 20 rollouts per task, yielding 80 substage evaluations per task to measure intermediate progress. Figure 4 compares spatial-granularity assignments to the two interfaces. Using the compact contextual tactile representation for context reasoning and the spatially dense tactile representation for action refinement achieves 85.0% average stage success, versus 73.8% for uniformly dense and 79.7% for uniformly compact representations, with the highest score on every task. This supports compact representations for contextual reasoning and spatial detail for action refinement. Removing contact-state classification loss reduces average stage success to 73.1%, indicating that contact supervision also contributes to representation learning.

4.6 Generalization to Appearance Changes

Figure 5 evaluates pen capping with an unseen pen color or a changed background, altering visual appearance while preserving the underlying contact interaction. The tactile observations remain structurally similar across these settings, providing stable contact information despite visual changes. TCR-ARM achieves 85% success in both settings, compared with 90% in the original setting. Dream-Tac drops from 80% to 65% and 70%, leaving gaps of 20 and 15 percentage points, respectively. These results demonstrate generalization to visual appearance changes when contact interaction remains similar.

4.7 Analysis of Tactile Context Reasoning and Action Refinement

Figure 6 examines the complementary roles of tactile context and gated refinement. On the left, each example pairs an image with V-JEPA teacher features (GT) and future predictions without and

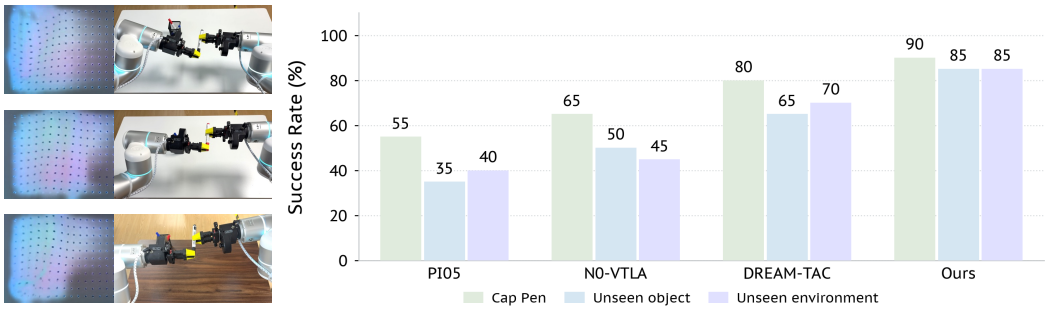

Figure 5: Generalization to appearance changes in pen capping. Left: visual and tactile observations. Right: success rates in the original and modified settings.

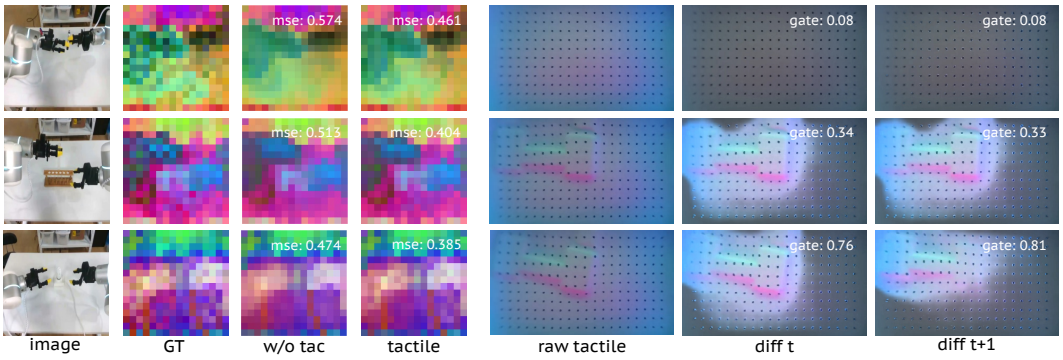

Figure 6: Left: image, V-JEPA GT, and future predictions without/with tactile modulation. Right: raw tactile images and difference maps at t and $t+1$ with gate values; rows show no contact, static contact, and dynamic contact.

with tactile modulation. All feature maps share the same PCA projection and color normalization, while the MSE to GT is shown for direct comparison. Tactile modulation consistently reduces prediction error across the examples, supporting the compact contextual tactile representation's role in improving predictive context for tactile context reasoning.

On the right, tactile observations and difference maps at t and $t+1$ illustrate no-contact, static-contact, and dynamic-contact conditions. The displayed gate values are low without contact (0.08/0.08), moderate and stable during static contact (0.34/0.33), and higher during dynamic contact (0.76/0.81). This pattern supports selective weighting of tactile updates according to interaction conditions, with stronger action refinement during changing contact. The compact context route remains available independently of this gate, preserving contact-state information even when local updates are attenuated. Together, the panels support persistent tactile context reasoning alongside selective local action refinement.

5 Conclusion

We present TCR-ARM, a VLA policy that combines persistent tactile context reasoning with selective action refinement. A shared encoder supplies a compact contextual tactile representation to the VLM and a spatially dense tactile representation for gated wrist-token updates in the action expert. RAT trains tactile context reasoning with and without tactile-driven action refinement. Across six contact-rich tasks, TCR-ARM achieves 70.0% average success, exceeding the strongest evaluated baseline by 15.0 percentage points. Ablations support the complementary roles of both interfaces and spatial-granularity assignment, with RAT adding 10.8 percentage points over full-tactile-only training. Our results support matching tactile representations to the needs of contextual reasoning and action refinement, and coordinating their use through route-aware training.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, including grammar correction, sentence refinement, and improving the clarity and organization of the manuscript. They were also used for suggestions regarding presentation and manuscript structure. All research ideas, methodological decisions, experimental design, implementation, data analysis, and scientific conclusions were developed and verified by the authors. All AI-assisted content was reviewed by the authors, who take full responsibility for the final manuscript.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the model architecture, training procedure, tactile data collection, post-training setup, and evaluation protocol in the main paper and supplementary material. Implementation details, hyperparameters, task configurations, and additional experimental results are provided in the appendix. These materials are intended to enable reproduction of the training and real-world evaluation procedures reported in this work.

REFERENCES

Jianxin Bi, Kevin Yuchen Ma, Ce Hao, Mike Zheng Shou, and Harold Soh. VLA-Touch: Enhancing Vision-Language-Action Model With Dual-Level Tactile Feedback. *IEEE Robotics and Automation Letters*, 11(7):8487–8494, 2026. doi: 10.1109/LRA.2026.3692345.

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. In *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40. PMLR, 2025a.

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. In *Proceedings of Robotics: Science and Systems*, 2025b. doi: 10.15607/RSS.2025.XXI.010.

Kevin Black, Allen Z. Ren, Michael Equi, and Sergey Levine. Training-Time Action Conditioning for Efficient Real-Time Chunking. arXiv preprint arXiv:2512.05964v2, 2025c.

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*, 11(9):10146–10153, 2026. doi: 10.1109/LRA.2026.3710060.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. In *Proceedings of Robotics: Science and Systems*, 2023. doi: 10.15607/RSS.2023.XIX.026.

Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal Manipulation Interface: In-The-Wild Robot Teaching Without In-The-Wild Robots. In *Proceedings of Robotics: Science and Systems*, 2024. doi: 10.15607/rss.2024.xx.045.

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.

Carolina Higuera, Akash Sharma, Chaithanya Krishna Bodduluri, Taosha Fan, Patrick Lancaster, Mrinal Kalakrishnan, Michael Kaess, Byron Boots, Mike Lambeta, Tingfan Wu, and Mustafa Mukadam. Sparsh: Self-supervised touch representations for vision-based tactile sensing. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 885–915. PMLR, 2025.

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:2507.09160v1, 2025.

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. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 2025.

Michelle A. Lee, Yuke Zhu, Krishnan Srinivasan, Parth Shah, Silvio Savarese, Fei-Fei Li, Animesh Garg, and Jeannette Bohg. Making Sense of Vision and Touch: Self-Supervised Learning of Multimodal Representations for Contact-Rich Tasks. In *IEEE International Conference on Robotics and Automation*, pp. 8943–8950, 2019. doi: 10.1109/ICRA.2019.8793485.

Xiaoqi Li, Muhe Cai, Jiadong Xu, Juan Zhu, Hongwei Fan, Yan Shen, Guangrui Ren, and Hao Dong. AT-VLA: Adaptive Tactile Injection for Enhanced Feedback Reaction in Vision-Language-Action Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28764–28774, 2026.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow Matching for Generative Modeling. In *International Conference on Learning Representations*, 2023.

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:2606.08737v1, 2026.

Siyu Ma, Yuqi Liang, Chang Yu, Yunuo Chen, Hao Su, Yixin Zhu, Yin Yang, and Chenfanfu Jiang. TacCoRL: Integrating Tactile Feedback into VLA via Simulation. arXiv preprint arXiv:2606.11743v1, 2026.

Charlotte Morissette, Amin Abyaneh, Wei-Di Chang, Anas Houssaini, David Meger, Hsiu-Chin Lin, Jonathan Tremblay, and Gregory Dudek. Tactile Modality Fusion for Vision-Language-Action Models. In *Computer Vision – ECCV 2026*, volume 17060 of *Lecture Notes in Computer Science*, pp. 21–40. Springer Nature Switzerland, 2026. doi: 10.1007/978-3-032-37092-1_2.

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 *Computer Vision – ECCV 2026*, volume 17005 of *Lecture Notes in Computer Science*, pp. 671–689. Springer Nature Switzerland, 2026. doi: 10.1007/978-3-032-37369-4_36.

NeoteAI Team and Fudan TEAI Team. N_0-TWAM: Scaling Tactile-Native World-Action Model for Contact-Rich Manipulation. arXiv preprint arXiv:2607.23783v1, 2026a.

NeoteAI Team and Fudan TEAI Team. N_0-VTLA: Scaling Vision-Tactile-Language-Action Model with Latent Tactile Tokens. arXiv preprint arXiv:2607.23782v1, 2026b.

Natalia Neverova, Christian Wolf, Graham W. Taylor, and Florian Nebout. ModDrop: Adaptive Multi-Modal Gesture Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(8):1692–1706, 2016. doi: 10.1109/tpami.2015.2461544.

Dantong Niu, Zhuoyang Liu, Zekai Wang, Boning Shao, Zhao-Heng Yin, Anirudh Pai, Yuvan Sharma, Stefano Saravalle, Ruijie Zheng, Jing Wang, Ryan Punamiya, Mengda Xu, Yuqi Xie, Yunfan Jiang, Letian Fu, Konstantinos Kallidromitis, Matteo Gioia, Junyi Zhang, Jiaxin Ge, Haiwen Feng, Fabio Galasso, Wei Zhan, David M. Chan, Yutong Bai, Roei Herzig, Jiahui Lei, Fei-Fei

Li, Ken Goldberg, Jitendra Malik, Pieter Abbeel, Yuke Zhu, Danfei Xu, Linxi Fan, and Trevor Darrell. T-Rex: Tactile-Reactive Dexterous Manipulation. arXiv preprint arXiv:2606.17055v2, 2026.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual Reasoning with a General Conditioning Layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pp. 3942–3951, 2018. doi: 10.1609/aaai.v32i1.11671.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856.

Jingwen Sun, Wenyao Zhang, Zekun Qi, Shaojie Ren, Zezhi Liu, Hanxin Zhu, Guangzhong Sun, Xin Jin, and Zhibo Chen. VLA-JEPA: Enhancing Vision-Language-Action Model with Latent World Model. In *Computer Vision – ECCV 2026*, volume 17064 of *Lecture Notes in Computer Science*, pp. 478–497. Springer Nature Switzerland, 2026. doi: 10.1007/978-3-032-37196-6_26.

Yi Wang, Renjun Wu, Jinyan Liu, and Xuesong Li. ViTaR: Visuo-Tactile Residual Adaptation for Foundation VLA Manipulation. arXiv preprint arXiv:2608.15816v1, 2026.

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*, 2025. doi: 10.15607/RSS.2025.XXI.052.

Fengyu Yang, Chenyang Ma, Jiacheng Zhang, Jing Zhu, Wenzhen Yuan, and Andrew Owens. Touch and Go: Learning from Human-Collected Vision and Touch. In *Advances in Neural Information Processing Systems*, volume 35, pp. 8081–8103, 2022. doi: 10.52202/068431-0587.

Fengyu Yang, Chao Feng, Ziyang Chen, Hyoungseob Park, Daniel Wang, Yiming Dou, Ziyao Zeng, Xien Chen, Rit Gangopadhyay, Andrew Owens, and Alex Wong. Binding Touch to Everything: Learning Unified Multimodal Tactile Representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26340–26353, 2024.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-WAM: Do World Action Models Need Test-time Future Imagination? arXiv preprint arXiv:2603.16666v2, 2026.

Wenzhen Yuan, Siyuan Dong, and Edward H. Adelson. GelSight: High-Resolution Robot Tactile Sensors for Estimating Geometry and Force. *Sensors*, 17(12):2762, 2017. doi: 10.3390/s17122762.

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:2603.12665v4, 2026a.

Pengwei Zhang, Bin Xie, Xinpan Meng, Xinyu Guo, Ce Hao, Fang Deng, Long Cheng, and Tiancai Wang. Feeling the Unexpected: ResTacVLA for Contact-Rich Manipulation via Residual Tactile Representation. arXiv preprint arXiv:2607.03387v3, 2026b.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. In *Proceedings of Robotics: Science and Systems*, 2023. doi: 10.15607/rss.2023.xix.016.

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. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183. PMLR, 2023.

A Dataset and Observation Examples

A.1 Data Composition and Visual Coverage

The pretraining corpus described in Section 3.4 contains 1,500 hours of tactile manipulation data across 128 tasks, including 1,000 hours collected with UMI and 500 hours collected with robots. The tasks are organized into three real-world scene families: *Office*, *Home*, and *Workbench*. Each family is further divided into tasks covering representative objects, manipulation primitives, and contact patterns of the corresponding environment. Office tasks include desk tidying, document and stationery handling, electronic devices, pantry interaction, and cable management; Home tasks cover kitchen and dining, food and drink handling, storage, cleaning, textiles, and personal care; Workbench tasks focus on precision assembly, tool operation, disassembly and repair, and inspection and sorting. Together, these tasks span diverse behaviors such as grasping, repositioning, insertion, alignment, tool–object interaction, and surface contact. Post-training uses 100 tactile demonstrations for each of the six evaluation tasks.

Figure 7 shows representative examples from these three scene families and both collection settings. The head view provides global workspace context, while the wrist views capture local object geometry and contact regions. Representative cases include object organization, container manipulation, precision assembly, tool interaction, and wiping, illustrating the diversity of objects, scenes, and contact-rich interactions covered by the pretraining corpus.



Figure 7: **Examples from the tactile manipulation dataset.** The left three columns show UMI collection and the right three columns show robot collection. Within each group, the columns provide a head view and two wrist views. Rows show different manipulation examples; corresponding rows across the two collection settings need not depict the same task.

A.2 Raw Tactile Observations and Input Construction

Figure 8 presents raw tactile images from the dataset. The images contain a visible marker pattern and spatially varying appearance under contact. Some observations retain a largely regular pattern, whereas others exhibit localized deformation and changes in the contact region. These examples motivate preserving spatial information for local refinement while forming a compact representation for contextual reasoning.

For the robot configuration, two tactile images on each gripper are concatenated vertically to form one logical tactile image per side. Each logical image is resized to 128×128. The encoder receives four frames per side in chronological order, at offsets $(-3, -2, -1, 0)$ from the current observation. Pixel values are mapped to $[0, 1]$ and normalized using channel means $(0.485, 0.456, 0.406)$ and standard deviations $(0.229, 0.224, 0.225)$. Frame and patch masks distinguish an unavailable observation from a valid observation with no contact.

Contact-label construction. Contact labels are derived from differences between a tactile image and a static reference image. A change exceeding a 5% threshold is labeled as contact; otherwise, the image is labeled as no contact. These binary labels supervise the contact head of the shared tactile encoder. They provide an auxiliary training signal and do not directly set the wrist-conditioned refinement gate. Missing labels are excluded from the contact loss.

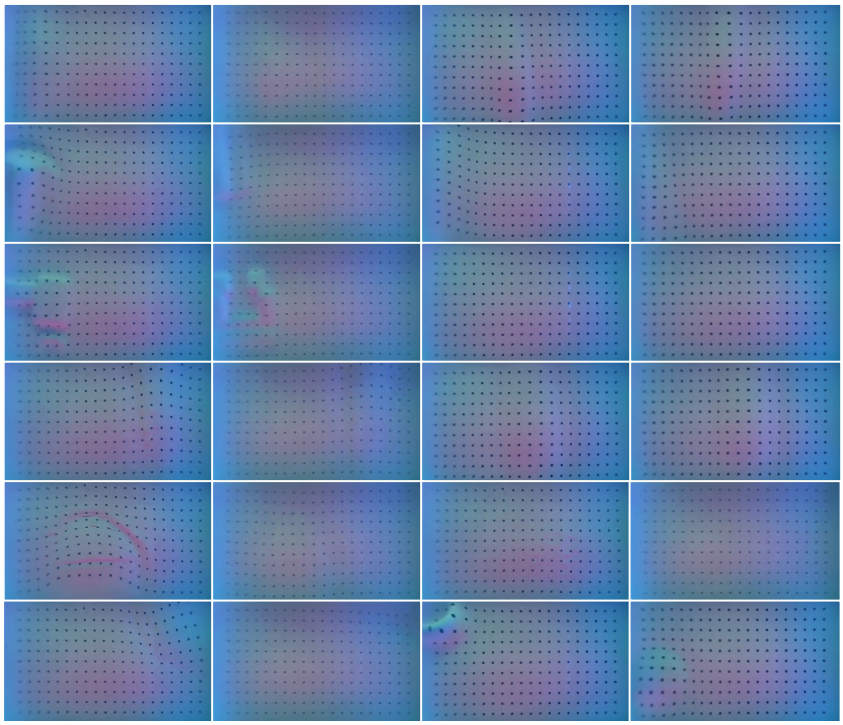

Figure 8: **Raw tactile examples.** The montage shows variation in marker patterns and local contact appearance across collected observations. These are sensor images, rather than predicted force fields or calibrated pressure maps. The panel illustrates tactile diversity without assigning a temporal ordering to its cells.

B Implementation Details

B.1 Representations and Action Encoding

Table 4 summarizes the implementation. One shared DINOv3 ViT-B/16 encoder processes both sides and all four tactile frames. Its class and register tokens are excluded from the tactile patch representations. Current-frame queries attend to the valid history independently for each side. Masked 2×2 spatial pooling and two Transformer blocks then produce compact contextual tokens. Separate linear projections map the compact and dense features to the VLM and action-expert widths, respectively.

Table 4: **Model and representation settings.** Token counts are reported across both tactile sides unless stated otherwise.

Setting	Value
Robot platform	Flexiv Rizon4 robot arms
VLM backbone	QWen3.5-2B
Action expert	18 layers; hidden width 1,024; MLP width 3,072
Action-expert topology	Six blocks, each with two linear-attention layers and one full-attention layer
Tactile backbone	DINOv3 ViT-B/16; feature width 768; jointly optimized
Tactile input	128×128; four frames per side; stride 1
Dense tactile representation	8×8 tokens per side; 128 total
Compact tactile representation	4×4 tokens per side; 32 total
Temporal / context attention	12 heads; two compact-context Transformer blocks
Refinement initialization	Zero-initialized attention output projection
Robot action / state	20 valid dimensions in a 34-slot implementation layout
Action horizon	40 steps
Future teacher	Frozen V-JEPA 2.1 ViT-g/16 target encoder
Future clip	Four 256×256 head-camera frames
Future frame offsets	$+10, +20, +30, +40$ from the current observation
Teacher token grid	$2 \times 16 \times 16$; 512 tokens of width 1,408
Future task / output tokens	16 VLM task tokens; 512 prediction queries
Prediction head	One layer; eight heads of dimension 128

Physical actions and implementation slots. The 20-dimensional action in Section 3.1 contains two end-effector poses (three position and six rotation coordinates each) and two gripper values. The shared data interface reserves 34 slots: 14 for arm joints, 18 for end-effector poses, and two for grippers. For the evaluated robot mapping, the arm-joint slots are unused and masked out of the loss; the valid 20 components are selected when converting outputs back to robot actions. Pose targets are expressed relative to the current state, while gripper targets remain absolute. Pose values use the dataset's 1st and 99th percentiles for normalization, followed by clipping to $[-1.5, 1.5]$; gripper values use min–max normalization.

Future-target geometry. The teacher uses spatial patches of size 16×16 and temporal tubelets of two frames. Four 256×256 frames therefore yield two temporal grids of 16×16 tokens. GT denotes the teacher encoding of an observed future clip. Future clips containing padded frames are excluded from the auxiliary loss.

B.2 Optimization, Route Sampling, and Deployment

Table 5 reports the supplied pretraining and post-training configuration values. Learning-rate groups distinguish the pretrained VLM, the tactile encoder, and the remaining trainable modules. Both visual and tactile encoders are trainable; the future teacher is frozen.

Table 5: **Optimization settings in the training configurations.** The microbatch size is per data-parallel worker; the global batch additionally depends on the number of workers. Contact supervision contributes only when valid contact labels are supplied.

Setting	Pretraining	Post-training
Optimizer	AdamW	AdamW
Adam (β_1, β_2); ϵ	$(0.9, 0.95)$; 10^{-8}	$(0.9, 0.95)$; 10^{-8}
VLM learning rate	10^{-6}	10^{-6}
Tactile / other learning rates	5×10^{-5}	5×10^{-5}
Schedule / warmup	Constant / none	Constant / none
Weight decay / gradient-norm cap	0 / 1.0	0 / 1.0
Microbatch / accumulation steps	240 / 1	240 / 1
Future loss coefficient λ_f	0.01	0.01
Contact loss coefficient λ_c	0.001	0.001
Tactile routes	Uniform RAT	Full tactile $(1, 1)$
RTC prefix conditioning	Disabled	Enabled; at most 8 steps
Numerical execution	BF16 autocast	BF16 autocast
Gradient checkpointing	Enabled	Enabled

Route masks and sensor validity. RAT samples one route independently for each training example, with probability $1/3$ for each of $(0, 0)$, $(1, 0)$, and $(1, 1)$. The sampled masks control whether encoded touch reaches the context and refinement interfaces. Sensor-validity masks instead identify which observations are usable. Contact classification reads the encoder representation before route masking, so a no-tactile route does not discard available contact supervision. Missing contact labels are masked rather than treated as negative labels. Post-training and inference enable both tactile routes, while the learned wrist-conditioned gate continues to weight the dense updates.

Action and auxiliary losses. The action flow time is sampled as $\tau = 0.999u + 0.001$, where $u \sim \mathrm{Beta}(1.5, 1.0)$. The configured future objective is patchwise MSE; cosine and smooth-L_1 losses are disabled. Gradients from this objective do not enter the teacher, and head-image hidden states are detached at the prediction-head input. Contextual tactile and future task hidden states retain their gradient paths.

RTC conditioning and inference. During post-training, a prefix of demonstrated actions remains clean while the suffix receives flow-matching noise and action supervision. For prefix lengths $d = 0, \ldots, 8$, the configured sampling weights are

$$(0.10,\ 0.02,\ 0.02,\ 0.02,\ 0.02,\ 0.02,\ 0.35,\ 0.35,\ 0.10).$$

Infeasible lengths are masked and the remaining weights are renormalized per example, leaving at least one valid suffix action. At inference, each request encodes the observation prefix once and reuses its attention keys and values across five Euler steps. Dense tactile refinement is enabled at all five steps. RTC overlaps action execution with generation of the next chunk. The contact classifier, future prediction head, and teacher are inactive during deployment.

C Evaluation Protocol and Stage Definitions

C.1 Full-Task Success

Every method is evaluated over 20 trials per task and condition, with a three-minute time limit per trial. Success requires completing all criteria in Table 6 within this limit. Partial completion or continued attempts without reaching the final criterion count as failure. For task k, the success rate is $100n_k/20$, where n_k is the number of successful trials. The six-task mean equally weights the task rates, and individual task rates change in increments of five percentage points.

Table 6: **Full-task success criteria.** Each criterion must be satisfied within three minutes.

Task	Success criterion
Wipe Plate	Remove the stains from the plate and place the plate back down.
Attach Pipette Tip	Attach the tip securely to the pipette without it falling off, then place the pipette down.
Insert Optical Module	Insert the optical module and press it firmly into place.
Cap Pen	Fit the cap onto the pen and place the pen down.
Insert Test Tube	Insert the test tube into the rack so that it remains upright rather than tilted.
Screw Light Bulb	Screw the bulb tightly into the socket and make it illuminate.

C.2 Four-Stage Progress Metric

The spatial-granularity study in Section 4.5 uses the ordered stages in Table 7. The 20 rollouts of each task yield $20 \times 4 = 80$ stage outcomes. A stage receives credit when its stated subtask is completed; an unreached or incomplete stage receives no credit. These outcomes measure progress within a rollout and are not 80 independent full-task trials.

Table 7: **Stage definitions for the spatial-granularity study.** Stages follow task execution order. "Tip" denotes the detachable pipette tip.

Task	Stage 1	Stage 2	Stage 3	Stage 4
Insert Optical Module	Pick up the module	Align it with the slot	Insert the module	Press it firmly into place
Attach Pipette Tip	Pick up the pipette	Pick up the tip	Attach the tip to the pipette	Place the assembled pipette down
Cap Pen	Pick up the cap	Pick up the pen	Fit the cap onto the pen	Place the capped pen down
Wipe Plate	Pick up the plate	Pick up the sponge	Wipe the plate	Place the plate down

Let $z_{kij} \in \{0, 1\}$ indicate completion of stage j in rollout i of task k. The task bar and the final average in Figure 4 are

$$S_k = \frac{100}{20 \times 4} \sum_{i=1}^{20} \sum_{j=1}^{4} z_{kij}, \qquad S_{\text{avg}} = \frac{1}{4} \sum_{k=1}^{4} S_k. \tag{10}$$

A rollout may complete early stages yet fail the full task. Thus, the 85.0% mean stage score and the 70.0% six-task completion rate summarize different outcomes and task sets. Rates are aggregated before rounding to one decimal place.

D Additional Future-Latent Visualizations

Figure 9 provides twelve qualitative examples across six tasks, complementing Section 4.7. Each example contains an original head-camera image (**Image**), teacher target features (**GT**), predicted features (**Predict**), and the prediction overlaid on the original image (**Predict***). Following the future-target definition in Section 3.4, GT is obtained by applying the frozen V-JEPA target encoder to recorded future frames. Predict is the policy's estimate of the corresponding latent representation.

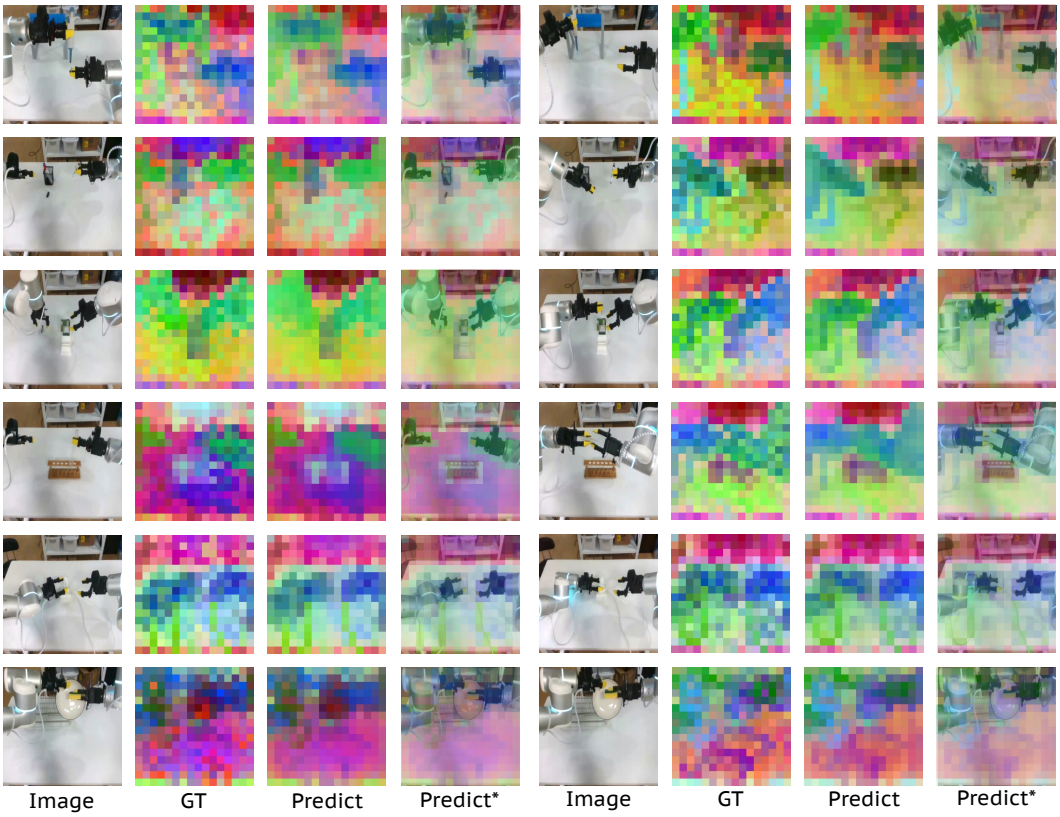

Figure 9: **Additional future-latent visualizations.** Each row contains two examples, each ordered as Image, GT, Predict, and Predict*. From top to bottom, the tasks are Attach Pipette Tip, Cap Pen, Insert Optical Module, Insert Test Tube, Screw Light Bulb, and Wipe Plate. GT denotes teacher features extracted from future observations. Predict* overlays the predicted feature map on the original image to help inspect its spatial correspondence with the scene.

Spatial and semantic interpretation. The color-projected features retain broad spatial organization: regions associated with the manipulators, manipulated objects, and background occupy distinguishable parts of the map. Predict reproduces much of this structure while exhibiting local differences from GT in color and region boundaries. The overlays make it easier to associate feature regions with visible scene content, illustrating how the predicted latent representation preserves spatially organized information relevant to manipulation.

These panels visualize latent features rather than decoded future RGB images. Predict* is a display overlay, not an additional prediction branch. The colors are not semantic class labels, and visual correspondence alone is not a semantic-segmentation score or a guarantee of successful execution. The examples complement the quantitative task results by showing the structure of the auxiliary prediction output.

E FAILURE CASE: OPTICAL-MODULE INSERTION

Figure 10 shows an unsuccessful optical-module insertion. The top row summarizes the overall attempt, while the second row enlarges repeated local attempts near the insertion interface. The final two rows show tactile difference images from the two sensors on the left gripper during these attempts. The tactile patterns change as the gripper continues interacting with the module and its surroundings.

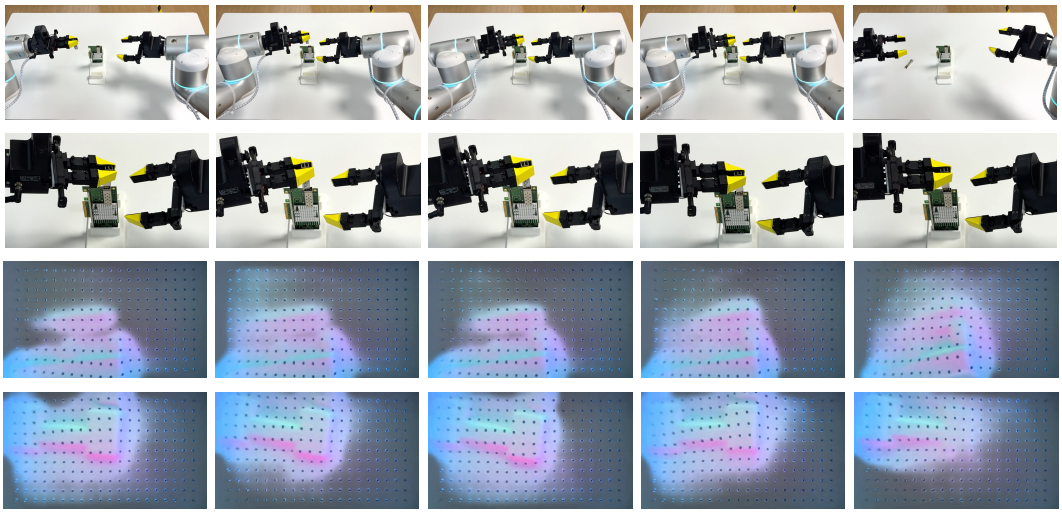

Figure 10: **Failure during optical-module insertion.** Top: the overall failed execution. Second row: local views of repeated insertion attempts. Third and fourth rows: tactile difference images from the two sensors on the left gripper during local exploration. The changing tactile patterns accompany continued interaction, but the execution does not meet the full-task success criterion.

Continued interaction without successful completion. The policy continues attempting local adjustments instead of immediately terminating after an unsuccessful insertion attempt. The accompanying changes in tactile observations are consistent with continued contact feedback during these adjustments. Nevertheless, the module is not successfully inserted and pressed into place, so the rollout remains a failure under Table 6. Repeated attempts within one rollout do not increase the number of evaluation trials or convert partial progress into task success.

Implications for recovery. This case illustrates that continued contact-sensitive behavior can coexist with unsuccessful task completion. It is consistent with the intended role of dense tactile refinement, but the image sequence alone does not isolate which tactile route caused a particular motion or identify the gate value at each instant. The case motivates improving recovery from unsuccessful engagement: local adjustments must eventually lead to a configuration that satisfies the insertion and seating criteria. Persistent tactile context and adaptive refinement provide useful feedback pathways, but do not by themselves guarantee recovery from every contact configuration.