TACMIND: BENCHMARKING AND MODELING TACTILE INTERACTION STATES FOR ROBOTIC MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Tactile sensing provides robots with direct observations of contact physics, local geometry, and interaction dynamics, making it a vital source of feedback for contact-rich manipulation. However, existing tactile manipulation studies are often developed and evaluated within specific task settings, making it difficult to systematically compare how tactile information contributes under different interaction conditions. In this work, we develop a benchmark in simulation comprising 128 tasks organized around three complementary evaluation regimes: Precise, Recognition, and Dynamic. The benchmark further includes controlled static and dynamic task variants and 12 perturbation settings for evaluating robustness under changes in visual and physical conditions. These settings motivate studying tactile information as a temporally evolving interaction representation rather than only as an instantaneous observation. We therefore propose an action-conditioned latent predictive world model that integrates tactile history and learns a tactile state through prediction of future tactile latents. Building on this representation, we design a model that conditions action generation on the learned tactile state and uses predicted future tactile information to refine candidate actions through predictive tactile reflection. Experiments on the proposed benchmark show consistent gains from incorporating tactile information across the evaluated regimes.

1 INTRODUCTION

Vision provides rich global information for robotic manipulation, while tactile sensing provides direct measurements of local physical interaction including contact geometry, deformation, slip, and object-dependent physical cues. Such information is useful not only when vision is ambiguous, but also for regulating contact and tracking interaction dynamics during manipulation Howe (1993); Hogan et al. (2020); Wu et al. (2025). Tactile sensing complements vision by directly observing local physical interactions, and recent visuo-tactile methods have shown benefits in contact-rich manipulation when visual observations are ambiguous or insufficient Luu et al. (2025); Chen et al. (2026); Zheng et al. (2026). However, tactile feedback is not only useful at the current timestep. During manipulation, it provides information about *what* is being contacted, *how* the interaction evolves, and how subsequent actions may affect that interaction. This motivates representing touch as an evolving interaction state rather than only as an instantaneous auxiliary observation.

Existing evaluation settings do not yet systematically characterize this broader role of touch. Many visuo-tactile methods are evaluated in real-world setups that are difficult to reproduce across laboratories, while recent simulation benchmarks have mainly studied capabilities such as high-precision manipulation and tactile object recognition Li et al. (2024); Luu et al. (2025); Chen et al. (2026). Precision tasks are often initialized from favorable contact states, such as with an object already grasped or placed near the target, and many contact-rich tasks use configurations that remain fixed during execution. These settings are useful for evaluating individual tactile skills, but provide limited evidence about how tactile information contributes as an interaction changes over time. In particular, it remains unclear how touch supports precise control, resolves visual ambiguity, adapts to changing interactions, and provides complementary cues under variations in visual observations or object properties.

To study these settings systematically, we introduce a capability-oriented tactile benchmark TACMIND-BENCH with 28 standard tasks and 100 extra tasks. The benchmark evaluates tactile sensing under three complementary conditions: *(i)* precise manipulation, where small contact errors affect task success; *(ii)* recognition under ambiguity, where task-relevant information is dif-

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.

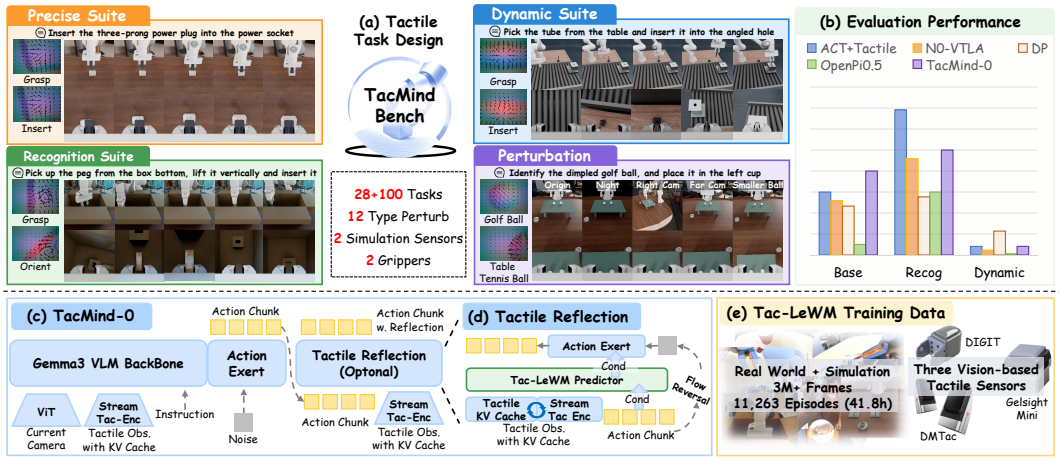

Figure 1: **Overview of TacMind-Bench and TacMind-0**. **(a)** TacMind-Bench comprises 28 standard and 100 additional tasks spanning precise contact, recognition under ambiguity, and dynamic interaction, together with controlled perturbations for robustness evaluation. **(c)** TacMind-0 is our VTLA model that maintains a streaming tactile state for action generation. **(d)** Predictive tactile reflection leverages Tac-LeWM to estimate future tactile states conditioned on candidate actions and uses them to refine action generation. **(e)** Tac-LeWM is pretrained on large-scale multi-sensor tactile manipulation demonstrations collected from both simulation and the real world. **(b)** TacMind-0 achieves competitive performance on TacMind-Bench Standard.

ficult to recover from vision alone; *(iii)* dynamic interaction, where the task state changes during execution together with controlled perturbations in lighting, camera configuration, and object properties. These settings place different demands on tactile representations. Recognition under visual ambiguity requires preserving local appearance, geometry, and object-dependent cues. Precise and dynamic manipulations additionally require information about contact deformation and its evolution over time. For control, the representation should also retain information that is useful for predicting how actions may change the physical interaction. We therefore consider three properties important for a general tactile state: *semantically informative*, *temporally grounded*, and *action-relevant*.

Learning such a state introduces several challenges. Vision-based tactile sensors provide complementary signals: tactile RGB observations capture local appearance and geometry, while marker deformation reflects interaction patterns such as shear, torsion, and sliding Luu et al. (2025). Tactile observations are also history-dependent, since similar instantaneous measurements may arise from different interaction trajectories. Finally, a representation useful for control should encode not only the current interaction but also information predictive of its subsequent evolution under action.

We address these challenges with TacMind-0, a vision-tactile-language-action framework that maintains a streaming tactile state throughout manipulation. Its tactile representation is learned by Tac-LeWM, a predictive latent model that integrates tactile appearance, deformation, and interaction history. Given a history of tactile observations and robot actions, Tac-LeWM predicts the latent representation of a future tactile chunk rather than reconstructing future tactile images. This joint-embedding objective encourages the current state to retain information predictive of subsequent interaction dynamics without requiring explicit reconstruction of potentially ambiguous sensory details. Our formulation is inspired by predictive latent world models Maes et al. (2026) and differs from approaches that primarily use tactile prediction for explicit future rollout or observation forecasting Higuera et al. (2026); Zang et al. (2026); here, future touch is used as supervision for learning the tactile representation.

Tac-LeWM further incorporates spatial and temporal structure specific to vision-based tactile sensing. A canonical tactile rotary positional encoding maps mirrored fingertip sensors into a shared local frame and encodes spatial relations along multiple directions, helping represent deformation patterns that are not aligned with the image axes. A streaming causal tactile encoder incrementally updates the tactile state while reusing cached historical features, allowing long tactile histories to be incorporated efficiently during online control.

After the tactile representation is obtained, our TacMind-0 uses the resulting state in two ways. First, the tactile state directly conditions action generation, providing physical information that may be ambiguous or unavailable from vision. Second, we introduce tactile reflection: after generating an initial action proposal, the model predicts the tactile latent state associated with that proposal and uses the predicted interaction state to refine the action. This allows the policy to use both

tactile evidence accumulated from previous interaction and information about the possible tactile consequence of a candidate action.

Our contributions are summarized as follows:

- We introduce a capability-oriented tactile benchmark with 28 standard tasks and 100 extra tasks covering precise manipulation, tactile recognition under visual ambiguity, dynamic interaction, and robustness under 12 types of controlled perturbations.
- We propose TacMind-0, a vision-tactile-language-action framework that maintains a streaming tactile state for action generation and refines candidate actions through predictive tactile reflection.
- We develop Tac-LeWM, a predictive tactile representation model that integrates tactile appearance, marker deformation, and long interaction histories through latent predictive learning, canonical spatial encoding, and streaming causal modeling.

2 Related Work

Tactile Manipulation Benchmarks. Recent tactile simulators and benchmarks have enabled increasingly systematic evaluation of tactile perception and contact-rich manipulation Wang et al. (2022); Si & Yuan (2022); Akinola et al. (2025); Nguyen et al. (2024); Li et al. (2024); Luu et al. (2025); Chen et al. (2026). Existing suites cover precision manipulation, tactile perception, representation comparison, and deformable-object interaction, with recent work further considering robustness and transferable tactile features Jing et al. (2026); Higuera et al. (2024). These benchmarks provide important evaluation infrastructure, but are typically organized around particular task families, sensing settings, or interaction regimes. Our benchmark instead organizes tasks by the role of touch in manipulation, covering precise control, visual ambiguity, dynamic interaction, and controlled perturbations within a common framework.

Tactile-Aware Manipulation and Predictive Modeling. Tactile feedback has been incorporated into manipulation through learned representations, reactive control, and predictive modeling. Prior work has studied transferable tactile representations Guzey et al. (2023); Xu et al. (2025); Higuera et al. (2024); Zhao et al. (2024); Higuera et al. (2025), visuo-tactile manipulation Yin et al. (2023); Yuan et al. (2024); Yu et al. (2023); Huang et al. (2024), and high-frequency tactile correction for contact-rich policies Xue et al. (2025); Bi et al. (2026); Zhang et al. (2026); Zheng et al. (2026). A closely related line models future tactile interaction for planning or control: VT-WM learns action-conditioned visuo-tactile dynamics, TacForeSight uses predicted tactile dynamics as an anticipatory control signal, and HiTac-WAM predicts structured contact consequences for candidate-action evaluation Higuera et al. (2026); Zang et al. (2026); Xue et al. (2026). Recent approaches such as TouchWorld and ViTacWorld further use predicted visuo-tactile trajectories for closed-loop refinement or rollout-based reasoning Zhou et al. (2026); Huang et al. (2026). Our Tac-LeWM instead uses future tactile latents as a learning signal for the current tactile state, which is then used by TacMind-0 for tactile-conditioned action generation and predictive tactile reflection.

3 TacMind-Bench Design

TacMind-Bench organizes manipulation tasks according to the dominant role of tactile information in task completion, rather than by object category or manipulation primitive. It consists of a **Standard** suite with 28 tasks and an **Extra** suite with 100 tasks. Both suites cover three evaluation regimes: **precise contact**, **recognition under ambiguity**, and **dynamic interaction**. TacMind-Bench Standard contains a curated set of representative tasks used for both training and evaluation. The Extra suite extends these tasks with greater diversity in textures and language descriptions, together with additional manipulation tasks involving new objects. Table 1 compares TacMind-Bench with representative tactile manipulation benchmarks. Unlike prior benchmarks that primarily focus on particular manipulation skills or tactile objectives, TacMind-Bench explicitly organizes evaluation around different roles of tactile information throughout manipulation.

Precise Contact. Precise contact tasks primarily evaluate precise insertion and mating under settings where neither the source object nor the target undergoes externally driven motion. Vision provides coarse spatial configuration between the object and target, but once contact is established, the local interaction state and the corrective actions required for precise manipulation can be difficult to infer from visual observations alone. These tasks therefore evaluate how tactile feedback can support fine contact regulation during contact-rich manipulation. Many existing precision-oriented tactile

Table 1: Comparison with representative tactile manipulation benchmarks. ✓ denotes an explicit benchmark design axis, △ denotes partial coverage, and – denotes that the dimension is not explicitly evaluated. **E2E** denotes manipulation that includes object acquisition rather than only the terminal contact stage. **SD.** denotes State Disambiguation. **Dyn.** denotes externally driven source or target motion during execution. **Perturb.** denotes an orthogonal controlled robustness axis.

Benchmark	Benchmark Objective	Tasks	E2E	SD.	Dyn.	Perturb.
ManiSkill-ViTac	Precision contact-rich manipulation	3	–	–	–	–
ManiFeel	Tactile-aware exploration and contact-rich manipulation	9	△	△	–	–
UniVTAC	Unified visuo-tactile manipulation	8	△	–	–	–
Tabero	Gentle manipulation with tactile force regulation	40	✓	–	–	–
TacMind-Bench	**Roles of touch throughout manipulation**	**128**	✓	✓	✓	✓

evaluations focus primarily on the terminal contact phase, often initializing the object in-hand or close to the target. Such settings provide a controlled evaluation of fine contact regulation, but exclude earlier stages such as object acquisition, transport, and the transition into contact. TacMind-Bench retains such contact-focused settings while additionally introducing tasks that begin from object pickup and proceed through transport, contact establishment, and final insertion or placement. This allows tactile feedback to be evaluated over a broader portion of the manipulation process.

Recognition. Recognition tasks require the robot to recover task-relevant physical information that is occluded, visually ambiguous, or difficult to observe from the available viewpoints. These tasks cover object pose, size, surface texture, texture orientation, and boundary geometry. For example, in the rectangular-peg task, the box partially occludes the object, and the wrist camera provides limited information about its in-hand pose. In another task, the robot must distinguish hexagonal prisms with different side-surface textures; although no explicit occlusion is introduced, the distant front camera cannot resolve the fine patterns well, while the top-down wrist view provides limited visibility of the side surfaces. Some prior tactile benchmarks include individual tasks involving tactile discrimination under limited visual observability. TacMind-Bench makes state disambiguation an explicit evaluation regime and extends these settings to partial occlusion, viewpoint limitations, low visual contrast between the target and its surroundings, and cluttered scenes with multiple distractors. Some tactile-search tasks further require the robot to accumulate and retain tactile evidence across multiple contacts before identifying the target. These settings evaluate richer tactile cues beyond contact force or marker deformation alone.

Dynamic Interaction. Dynamic-interaction tasks extend precision manipulation by introducing externally driven motion. We vary where this motion enters the manipulation process in three ways. First, in *moving-target alignment*, the object is already held by the gripper while the corresponding target moves, separating online target alignment from object acquisition. Second, *static-source to moving-target* tasks retain the complete pick-and-insert procedure while the target continuously changes position during execution. Third, *moving-source to static-target* tasks require the robot to acquire an object from a moving source before manipulating it toward a stationary target. These settings distinguish dynamics primarily affecting target alignment from those affecting object acquisition. Where possible, dynamic tasks are paired with static counterparts while preserving the underlying object, target, and manipulation primitive. This controlled pairing enables comparison between stationary and time-varying interaction conditions.

Controlled Perturbations. Following common evaluation protocols, we vary object positions across trials. In addition, TacMind-Bench introduces controlled perturbations to both visual observations and physical task conditions. These perturbations are evaluated as an axis orthogonal to three task regimes. Visual perturbations include changes in illumination and camera viewpoint. These perturbations modify the visual observations while leaving the tactile sensing conditions unchanged, enabling evaluation of the complementary contribution of touch when visual observations vary. We further perturb object size. Compared with changes in object position or visual appearance, moderate scale variations introduce subtle visual differences while substantially altering the contact geometry encountered during manipulation. Such variations are particularly relevant to precision tasks and tasks where tactile feedback is used to regulate grasp force or contact during execution.

4 TacMind-0

4.1 Overview

TacMind-0 is a vision-tactile-language-action model that maintains a predictive tactile interaction state throughout manipulation. At time t, the robot receives visual observations I_t, a language instruction L, and a tactile observation τ_t. We denote the tactile history as $\mathcal{H}_t^\tau = (\tau_{t-H+1}, \ldots, \tau_t)$,

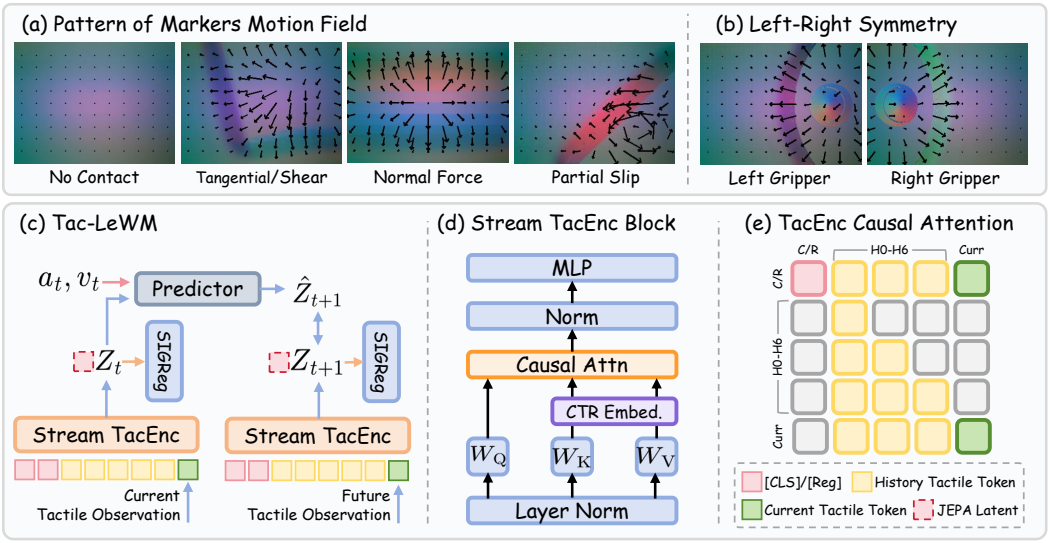

Figure 2: **Overview of Tac-LeWM. (a–b)** Marker-motion directions vary with contact conditions, while the left and right fingertip sensors have mirrored spatial layouts. These properties motivate tactile-specific spatial encoding that accounts for directional deformation and cross-fingertip symmetry. **(c)** Tac-LeWM learns a predictive tactile state by combining a streaming tactile encoder with an action-conditioned latent predictor trained to predict future tactile embeddings. **(d)** The streaming encoder reuses historical features through cached keys and values, enabling efficient online updates. **(e)** Causal attention integrates preceding tactile context while preventing access to future observations.

where H is the tactile context length. The policy predicts an action chunk $A_t = [a_t, \ldots, a_{t+K-1}]$, where K is the action horizon. Tac-LeWM encodes $\mathcal{H}_t^\tau$ into a streaming tactile state z_t^τ, learned through action-conditioned prediction of future tactile latents. TacMind-0 combines this state with vision and language to generate $A_t^{(0)}$, optionally refined through predictive tactile reflection:

$$\{I_t, \tau_t, L\} \xrightarrow{\pi_\theta} A_t^{(0)} \xrightarrow{P_\phi} \hat{Z}_t^+ \xrightarrow{\pi_\theta} A_t^{(*)}, \tag{1}$$

where P_ϕ is the Tac-LeWM predictor, and $\hat{Z}_t^+$ denotes the predicted future tactile latent.

4.2 Learning a Predictive Tactile State

Tac-LeWM adapts the joint-embedding predictive formulation of LeWM Maes et al. (2026) to vision-based tactile sensing, learning from future embeddings rather than reconstructing pixels. We modify the input representation and spatio-temporal encoding to account for tactile structure.

Tactile Input Representation. For each finger $f \in \{L, R\}$, we overlay the marker deformation representation $\tau_t^{f,\text{mark}}$ onto the tactile RGB image $\tau_t^{f,\text{rgb}}$ to obtain $\tilde{\tau}_t^f = \mathcal{O}(\tau_t^{f,\text{rgb}}, \tau_t^{f,\text{mark}})$, where $\mathcal{O}$ denotes the overlay operation. This combines local appearance and geometry with deformation cues such as shear, torsion, and sliding. The left and right fingertip observations are arranged into $\tau_t = \mathcal{M}(\tilde{\tau}_t^L, \tilde{\tau}_t^R)$ and patchified into tactile encoder tokens.

Canonical Tactile Rotary (CTR) Positional Embedding. Image positional encodings typically associate tokens with their locations on the image raster, while axial RoPE encodes horizontal and vertical offsets separately Heo et al. (2024); Liu et al. (2026). For tactile observations, however, the spatial structure of contact-induced deformation is not necessarily aligned with these axes. As illustrated in Fig. 2(a), shear, normal loading, and partial slip produce different marker-displacement fields whose directions can vary across the contact region. For example, normal loading can produce outward displacement, whereas partial slip produces nonuniform local motion. Interpreting these patterns involves relating deformation cues across spatial locations, rather than considering each marker displacement independently. This motivates a positional representation that explicitly encodes spatial relationships along multiple directions. The tactile input introduces an additional geometric consideration. Global raster coordinates depend on how the fingertip images are arranged in the input mosaic, while the left and right sensors are mounted in mirrored orientations, as shown in Fig. 2(b). We therefore use local fingertip coordinates rather than global mosaic positions. For a token from finger f at time t and local patch coordinate (y, x), we use $c_{f,t,y,x} = (t, y, x)$ and retain sensor identity through a separate learnable finger embedding. The right fingertip is

mapped into a shared canonical frame using $x^{\text{can}} = W_{\text{cell}} - 1 - x^{\text{raw}}$, where W_{cell} is the local patch-grid width. This mapping changes only positional coordinates, leaving the input image unchanged. Within this canonical frame, we project local coordinates along eight directions as $u_k(x, y) = x \cos \phi_k + y \sin \phi_k$, where $\phi_k = k\pi/8$ and $k \in \{0, \ldots, 7\}$. We encode these projections and the temporal index through rotary transformations of the query and key features in tactile causal attention. For two tokens separated by $(\Delta x, \Delta y)$, the corresponding spatial rotary phase differences depend on $\Delta u_k = \Delta x \cos \phi_k + \Delta y \sin \phi_k$. Each projection thus provides a relative-position cue along a particular orientation, including directions oblique to the image axes. The tactile features carry the observed deformation information, while CTR supplies a shared, direction-aware positional structure for learning spatial relationships among these features.

Streaming Causal Encoding. Tactile observations are history-dependent: similar instantaneous contact patterns may result from different preceding interactions. We therefore maintain the tactile state causally over time. Let $Z_t^\tau = E_\psi(\tau_t; \mathrm{KV}_{<t})$ denote the tactile token features at time t, where $\mathrm{KV}_{<t}$ contains cached keys and values from preceding tactile observations. Current tactile tokens may attend to current and historical observations, while future tactile observations are excluded from the online state. We use the encoder's global tactile token as the interaction state, denoted by $z_t^\tau = \mathrm{CLS}(Z_t^\tau)$. At inference, cached historical keys and values are reused when a new tactile observation arrives, allowing the tactile state to be updated incrementally without repeatedly encoding the complete history. The KV cache therefore serves as an efficient implementation of temporal context, while z_t^τ is the latent state consumed by the predictor and action policy.

Action-Conditioned Latent Prediction. For each time t, we define a future tactile chunk as $\mathcal{T}_t^+ = (\tau_{t+\Delta}, \ldots, \tau_{t+\Delta+C-1})$, where Δ is the prediction offset and C is the prediction horizon. The corresponding latent targets are $Z_t^+ = (z_{t+\Delta}^\tau, \ldots, z_{t+\Delta+C-1}^\tau)$. As in LeWM, the tactile encoder and predictor are optimized jointly, and the future embeddings are produced by the same tactile encoder rather than by a separate target network. Given the intervening actions $A_t^+ = (a_t, \ldots, a_{t+\Delta+C-2})$, the predictor estimates $\hat{Z}_t^+ = P_\phi\left(z_t^\tau, A_t^+, v_t\right)$, where v_t is an auxiliary visual feature used during Tac-LeWM pretraining and fine-tuning. A frozen DINOv3 Siméoni et al. (2025) encoder extracts v_t from the current scene observation to help disambiguate future tactile outcomes. Visual conditioning is randomly dropped during training and omitted at inference, while z_t^τ is computed only from tactile observations up to time t.

Predictive Learning Objective. Tac-LeWM predicts the future tactile embeddings rather than reconstructing future tactile images. The prediction objective is

$$\mathcal{L}_{\text{pred}} = \frac{1}{C} \sum_{j=0}^{C-1} \left\| \hat{z}_{t+\Delta+j}^\tau - z_{t+\Delta+j}^\tau \right\|_2^2. \tag{2}$$

Following LeWM Maes et al. (2026), we additionally apply SIGReg to reduce representation collapse. The complete Tac-LeWM objective is

$$\mathcal{L}_{\text{Tac-LeWM}} = \mathcal{L}_{\text{pred}} + \lambda \, \mathrm{SIGReg}(Z^\tau), \tag{3}$$

where λ controls the regularization strength. Both E_ψ and P_ϕ are optimized under this objective.

4.3 Tactile-Conditioned Action Generation

VTLA Architecture. TacMind-0 builds on a pretrained vision-language-action policy with a Gemma 3 vision-language backbone and a $\pi_{0.5}$-style flow-matching action expert. Tac-LeWM provides the streaming tactile state used by the policy, as shown in Fig. 1 (c). Given the visual observation I_t, language instruction L, and tactile state z_t^τ, we form the multimodal conditioning $c_t = \mathcal{F}(I_t, L, z_t^\tau)$. The tactile features are projected to the hidden dimension of the action expert and inserted as tactile conditioning tokens, while temporal tactile aggregation remains within the Tac-LeWM encoder. The policy then generates an initial action chunk as $A_t^{(0)} = \pi_\theta(c_t)$. Each action $a_t \in \mathbb{R}^8$ contains a 7-DoF arm command and one continuous gripper command. The action expert follows the standard conditional flow-matching formulation. Given a demonstrated action chunk A_t^{gt} and Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$, we interpolate them at flow time $s \in [0, 1]$ as $A_s = (1-s)\epsilon + sA_t^{\text{gt}}$, with target velocity $u_s = A_t^{\text{gt}} - \epsilon$. The action expert is trained using

$$\mathcal{L}_{\text{FM}} = \mathbb{E}\left[\left\| v_\theta(A_s, s \mid c_t) - u_s \right\|_2^2 \right]. \tag{4}$$

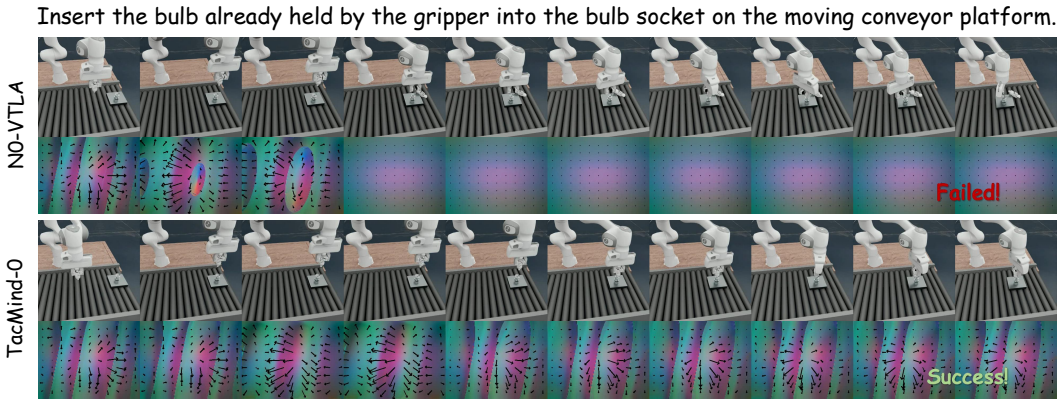

Figure 3: Comparision between N0-VTLA and TacMind-0 on the dynamic suit of TacMind-Bench.

At inference, integrating the velocity field from noise produces the initial action proposal $A_t^{(0)}$.

Predictive Tactile Reflection. The initial action policy uses tactile evidence accumulated up to the current time, but it does not explicitly condition action generation on the tactile state predicted under its own candidate action. We therefore introduce predictive tactile reflection. Given the initial proposal $A_t^{(0)}$, Tac-LeWM predicts the corresponding future tactile latent chunk as $\hat{Z}^+_{t,\mathrm{ref}} = P_\phi(z_t^\tau, A_t^{(0)})$. The auxiliary visual input used during Tac-LeWM pretraining is omitted here. We summarize the predicted tactile chunk as $\hat{z}^+_{t,\mathrm{ref}} = \mathrm{Pool}(\hat{Z}^+_{t,\mathrm{ref}})$ and use it directly as an additional action-conditioning signal rather than decoding it into tactile images. We implement action refinement through flow reversal steering. Let $F_\theta(\cdot \mid c)$ denote the flow transformation from latent noise to an action chunk. Starting from the initial action $A_t^{(0)}$, we reverse the base flow to obtain $\tilde{A}_0 = F_\theta^{-1}(A_t^{(0)} \mid c_t)$. We then augment the policy conditioning with the predicted tactile state, $c_t^{\mathrm{ref}} = \mathcal{G}(c_t, \hat{z}^+_{t,\mathrm{ref}})$, where $\mathcal{G}$ injects the predicted tactile feature into the flow-matching layers through adaptive normalization. The refined action is generated by $A_t^{(*)} = F_\theta(\tilde{A}_0 \mid c_t^{\mathrm{ref}})$. The reverse pass uses the original conditioning c_t, whereas the second forward pass additionally receives the predicted tactile state. The reverse–forward procedure therefore changes the conditioning under which the action is generated rather than simply reconstructing the initial proposal. The two tactile signals play distinct roles: (i) z_t^τ summarizes the observed interaction history and (ii) conditions the initial action, while $\hat{z}^+_{t,\mathrm{ref}}$ represents the tactile state predicted under the candidate action and conditions action refinement.

4.4 Training and Online Execution

Tac-LeWM Pretraining and Fine-tuning. For pretraining and fine-tuning, we jointly optimize the tactile encoder E_ψ and predictor P_ϕ using $\mathcal{L}_{\text{Tac-LeWM}}$. We also use action representation regularization in pretraining. The auxiliary DINOv3 Siméoni et al. (2025) feature is used only during this stage and is randomly dropped during training. Since the training datasets differ in their action representations, we do not enforce a unified preprocessing pipeline across all datasets. Instead, we use lightweight MLP projections to map actions from different datasets into a shared representation space, followed by a shared action encoder that extracts action features used to condition the predictor. More details about pretraining see Appendix.

VLA Post-Training. We perform two-stage post-training. First, the fine-tuned Tac-LeWM's encoder is frozen and used to provide z_t^τ to TacMind-0. The action expert is optimized using $\mathcal{L}_{\mathrm{FM}}$. Second, we freeze the tactile encoder, TacMind-0 VLM backbone, and action expert; only the predictor of Tac-LeWM and the AdaRMSNorm layer are trainable using action flow-matching loss and Tac-LeWM training loss.

Inference. At inference, future and history tactile observations are not required. The tactile encoder incrementally updates its KV cache as new tactile observations arrive. TacMind-0 first generates $A_t^{(0)}$ conditioned on (I_t, L, z_t^τ). When tactile reflection is enabled, Tac-LeWM predicts the future tactile latent from $A_t^{(0)}$, and a second conditioned flow pass produces the refined action $A_t^{(*)}$.

5 Experiments

5.1 Setup

We conduct experiments to evaluate our method and baseline models on the TacMind-Bench Standard benchmark. To provide a comprehensive evaluation, each task is evaluated for $(1+12) \times 20 =$

Table 2: Performance on TacMind-Bench (Gelsight Mini). OOD performance is categorized into lighting, camera, and size perturbations.

Model	Precise				Recognition				Dynamic				Avg	
	InD	Light	Cam	Size	InD	Light	Cam	Size	InD	Light	Cam	Size	InD	OOD
ACT+Tactile	30.0	26.0	2.5	2.5	69.1	48.9	31.7	13.9	4.1	1.9	0.2	0.2	34.4	14.2
DP	23.3	23.3	0.0	2.9	27.7	23.2	6.1	5.0	11.4	8.5	1.6	0.2	20.8	7.9
OpenPI0.5	5.0	0.0	0.0	0.0	30.0	24.2	1.2	15.9	0.5	0.2	0.0	0.2	11.8	4.6
OpenDM0.5	20.0	17.3	2.5	6.5	38.6	28.1	20.1	16.9	3.2	0.9	0.6	0.2	20.6	10.4
N0-VTLA	25.8	15.0	6.8	4.2	45.9	18.4	8.9	12.0	2.3	0.5	0.5	0.2	24.7	7.4
TacMind-0	40.0	35.0	14.4	17.9	50.0	43.6	35.0	28.1	4.1	2.6	0.2	0.5	31.4	19.7

Table 3: Performance on TacMind-Bench (DMTac). OOD performance is categorized into lighting, camera, and size perturbations.

Model	Precise				Recognition				Dynamic				Avg	
	InD	Light	Cam	Size	InD	Light	Cam	Size	InD	Light	Cam	Size	InD	OOD
ACT+Tactile	25.8	9.2	1.7	1.0	45.5	23.1	15.2	7.3	3.6	2.0	0.1	0.1	25.0	6.6
N0-VTLA	16.9	12.3	3.1	2.9	32.2	13.4	3.3	6.8	0.5	0.0	0.0	0.1	16.5	4.7
TacMind-0	23.8	16.3	6.8	4.4	38.8	28.7	20.3	15.6	3.1	0.9	0.1	0.2	21.9	10.4

260 episodes, covering one in-domain (InD) setting and 12 out-of-domain (OOD) settings with variations in factors such as lighting conditions, camera poses, and object sizes. Each setting is evaluated over 20 episodes. Therefore, evaluating all 28 standard tasks requires a total of 7,280 evaluation episodes per gripper. In addition, our benchmark includes two grippers equipped with different tactile sensors: the widely used GelSight Mini and the recently developed DMTac sensor. This setup requires substantially more evaluations than prior benchmarks, as models are evaluated across both tactile sensing configurations. We report the average success rates separately for the InD setting and the OOD settings.

Implementation Details. TacMind-0 is initialized from the pretrained OpenDM0.5 model, which adopts Gemma 3 as the VLM backbone and a $\pi_{0.5}$-style action expert. Tac-LeWM is pretrained on multiple tactile manipulation datasets collected from different sources. Further details of the Tac-LeWM pretraining procedure are provided in the Appendix. To ensure a fair comparison, TacMind-Bench is excluded from both the pretraining and fine-tuning stages of all models. For Gelsight Mini Gripper, we train the models using 100 demonstrations per task; for DMTac, we use 50 demonstrations per task. We train TacMind-0 and Tac-LeWM using 8 NVIDIA A100 GPUs. For TacMind-0 and the other baselines, we use an action chunk size of 32. For ACT-Tactile and Diffusion Policy, we train a separate model for each task, whereas the VLA and VTLA models are jointly trained across all tasks. For ACT-Tactile, we follow the model architecture of UniVTAC Chen et al. (2026). For Diffusion Policy Chi et al. (2025), we adopt the implementation provided by LeRobot Cadene et al. (2026). For OpenPI0.5 Intelligence et al. (2025), OpenDM0.5 Dexmal Team (2026), and N0-VTLA Team et al. (2026), we use their respective official codebases for training. For all models, we use relative joint actions as the action representation. We observe that training a policy on a single task is generally easier than jointly training it across the entire TacMind-Bench Standard task suite. Therefore, we do not recommend directly comparing the performance of single-task policies with that of multi-task policies, as they are trained under substantially different learning settings.

5.2 Evaluation Results

TacMind-Bench Results. The performance of our method and previous methods is reported in Table 2 and Table 3. Both the in-distribution (InD) and out-of-distribution (OOD) settings of TacMind-Bench are used for evaluation. The results show that our model outperforms previous tactile manipulation methods, especially with perturbation. The qualitative results are displayed in Figure 3, and more visual results are presented in the appendix.

Table 4: **Ablation study of TacMind-0.** We evaluate the contributions of tactile features and tactile reflection across different task settings and distributions.

Tactile	Reflection	Precise		Recon		Dynamic		Avg.	
		InD	OOD	InD	OOD	InD	OOD	InD	OOD
		20.0	8.8	38.6	21.7	3.2	0.6	20.6	10.4
✓		36.2	12.4	47.4	30.2	2.8	0.2	28.8	14.3
✓	✓	40.0	22.4	50.0	35.6	4.1	1.1	31.4	19.7

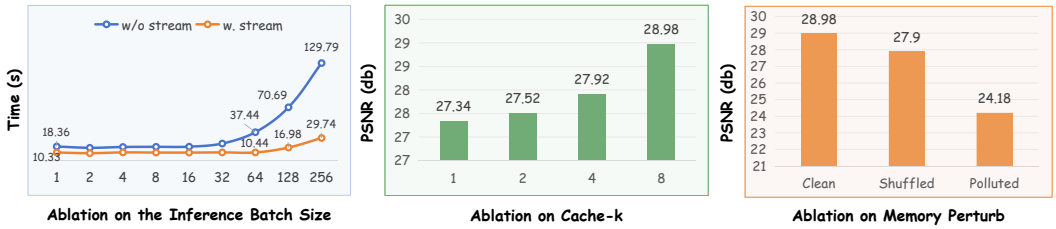

Figure 4: Ablation of Tac-LeWM on (a) stream inference, (b) length of memory, and (c) perturbing the memory.

Table 5: Ablation study on Tac-LeWM: linear probing results.

Variant	Proprio (MSE$_z$ ↓ / r ↑)	Gripper (MSE$_z$ ↓ / r ↑)	Action (MSE$_z$ ↓ / r ↑)	Pad (MSE$_z$ ↓ / r ↑)
Full	0.1816 / 0.8981	0.1958 / 0.9053	0.1841 / 0.8976	0.2771 / 0.7968
−Camera	0.1748 / 0.9027	0.1958 / 0.9053	0.1770 / 0.9022	0.2905 / 0.7855
−Causal Attn	0.1733 / 0.9035	0.1958 / 0.9053	0.1756 / 0.9030	0.3557 / 0.7282
−CTR	0.1831 / 0.8971	0.1958 / 0.9053	0.1851 / 0.8968	0.3366 / 0.7476

Table 6: Ablation study on Tac-LeWM: current and future reconstruction results.

	Current Reconstruction					Future Reconstruction				
Variant	PSNR↑	MSE↓	RMSE↓	MAE↓	SSIM↑	PSNR↑	MSE↓	RMSE↓	MAE↓	SSIM↑
Full	31.83	**0.00157**	**0.0341**	**0.0223**	**0.8748**	**28.98**	**0.00175**	**0.0391**	**0.0263**	**0.8593**
−Camera	**32.66**	0.00177	0.0348	0.0230	0.8649	26.52	0.00255	0.0489	0.0342	0.8034
−Causal Attn	31.45	0.00208	0.0385	0.0252	0.8451	23.07	0.00558	0.0724	0.0494	0.7116
−CTR	30.85	0.00190	0.0372	0.0242	0.8620	23.42	0.00567	0.0714	0.0518	0.7088

5.3 Ablation Study

TacMind-0. We ablate the key components of TacMind-0, as shown in Table 4. 'Tactile" denotes the incorporation of tactile features into the VLM backbone, while 'Reflection" refers to our proposed tactile reflection method. Incorporating tactile features substantially improves performance on both the Base and Recon settings, demonstrating the benefit of explicitly modeling tactile information. Further introducing tactile reflection yields consistent gains across nearly all settings, with notable improvements on OOD tasks, highlighting its effectiveness in enhancing generalization.

Tac-LeWM Components. We ablate three components of Tac-LeWM. −Camera removes the auxiliary visual conditioning from the predictor, −Causal Attn replaces causal attention with standard self-attention, and −CTR replaces Canonical Tactile RoPE with the original ViT positional embedding. As shown in Table 5, the variants perform similarly on most linear probes, while removing causal attention or CTR produces a clearer degradation in tactile pad prediction. This suggests that these designs have a larger effect on deformation-related tactile information than on linearly decoding proprioceptive or action variables. Table 6 shows a more pronounced difference in future reconstruction. While current reconstruction remains relatively similar across variants, future prediction degrades substantially without causal attention or CTR. These results suggest that causal temporal modeling and tactile-specific positional encoding are particularly useful for predicting future tactile representations, while visual conditioning provides additional predictive context.

Temporal Context and Streaming. We further examine the role of tactile history in Fig. 4. Increasing the cache length from $k = 1$ to $k = 8$ improves prediction PSNR from 27.34 dB to 28.98 dB, suggesting that preceding tactile observations provide useful context beyond the current frame. Perturbing this context reduces prediction quality, with PSNR decreasing to 27.90 dB after shuffling and to 24.18 dB under memory pollution. Finally, streaming inference reduces the cost of maintaining this history. At batch size 256, latency decreases from 129.79 ms to 29.74 ms by reusing cached historical features instead of recomputing the full context.

6 Conclusion

In this work, we design a tactile manipulation benchmark comprising of 28 standard and 100 additional tasks spanning precise contact, recognition under ambiguity, and dynamic interaction. To address the challenges in these tasks, we propose a predictive latent model and a vision-tactile-language-action model, achieving the best performance compared with previous methods.

AI USE STATEMENT

In this work, we used generative AI tools to assist with polishing and improving the clarity of the manuscript, debugging code, and supporting data collection for our benchmark. We did not use generative AI tools to independently formulate the main research claims or conclusions. All AI-assisted manuscript edits were reviewed by the authors. AI-assisted code debugging was followed by manual inspection and testing to verify correctness. For benchmark data collection, AI-generated or AI-assisted outputs were reviewed and filtered by the authors according to our data collection protocol and quality-control criteria. We take full responsibility for the final content of this work, including all text, claims, code, and benchmark data produced with the aid of generative AI.

ETHICS STATEMENT

Our benchmark is built in the simulation. The collected data will be released, but there is no human subjects data recorded.

REPRODUCIBILITY STATEMENT

The design of our benchmark is introduced in Section 3. The framework and training/inference setup of our model are introduced in Section 4. The details of the benchmark and experiment setup are stated in Section 5 and the appendix. The data and code of the benchmark and the model will be made public.

REFERENCES

Iretiayo Akinola, Jie Xu, Jan Carius, Dieter Fox, and Yashraj Narang. Tacsl: A library for visuotactile sensor simulation and learning. *IEEE Transactions on Robotics*, 41:2645–2661, 2025.

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*, 2026.

Remi Cadene, Simon Alibert, Francesco Capuano, Michel Aractingi, Adil Zouitine, Pepijn Kooijmans, Jade Choghari, Martino Russi, Caroline Pascal, Steven Palma, et al. Lerobot: An open-source library for end-to-end robot learning. In *International Conference on Learning Representations*, volume 2026, pp. 122398–122417, 2026.

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.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

Daimon Robotics. Daimon-infinity, 2026. URL `https://modelscope.cn/datasets/daimonrobotics/Daimon-Infinity`.

Dexmal Team. Opendm, 2026. URL `https://github.com/dexmal/opendm`.

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. URL `https://openreview.net/forum?id=ndilONnABZ`.

Tamar Flash and Neville Hogan. The coordination of arm movements: an experimentally confirmed mathematical model. *The journal of Neuroscience*, 5(7):1688–1703, 1985.

Irmak Guzey, Ben Evans, Soumith Chintala, and Lerrel Pinto. Dexterity from touch: Self-supervised pre-training of tactile representations with robotic play. *arXiv preprint arXiv:2303.12076*, 2023.

Byeongho Heo, Song Park, Dongyoon Han, and Sangdoo Yun. Rotary position embedding for vision transformer. In *European Conference on Computer Vision*, pp. 289–305. Springer, 2024.

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, pp. 885–915, 2024. URL https://mlanthology.org/corl/2024/higuera2024corl-sparsh/.

Carolina Higuera, Akash Sharma, Taosha Fan, Chaithanya Krishna Bodduluri, Byron Boots, Michael Kaess, Mike Lambeta, Tingfan Wu, Zixi Liu, Francois Robert Hogan, et al. Tactile beyond pixels: Multisensory touch representations for robot manipulation. In *Conference on Robot Learning*, pp. 105–123. PMLR, 2025.

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

Francois R Hogan, Jose Ballester, Siyuan Dong, and Alberto Rodriguez. Tactile dexterity: Manipulation primitives with tactile feedback. In *2020 IEEE international conference on robotics and automation (ICRA)*, pp. 8863–8869. IEEE, 2020.

Robert D Howe. Tactile sensing and control of robotic manipulation. *Advanced Robotics*, 8(3): 245–261, 1993.

Binghao Huang, Yixuan Wang, Xinyi Yang, Yiyue Luo, and Yunzhu Li. 3d-vitac: Learning fine-grained manipulation with visuo-tactile sensing. *arXiv preprint arXiv:2410.24091*, 2024.

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

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

Bowen Jing, Mingxin Wang, Ruiyang Hao, Chenchen Ge, Hanwen Shen, Junjie He, Yang Cui, Yiming Hou, Weitao Zhou, Jiawei Wang, et al. Softvtbench: A safety-aware visuo-tactile benchmark for physically constrained robotic manipulation of deformable objects. *arXiv preprint arXiv:2607.04234*, 2026.

Chuanyu Li, Renjun Dang, Xiang Li, Zhiyuan Wu, Jing Xu, Hamidreza Kasaei, Roberto Calandra, Nathan Lepora, Shan Luo, Hao Su, et al. Maniskill-vitac 2025: challenge on manipulation skill learning with vision and tactile sensing. *arXiv preprint arXiv:2411.12503*, 2024.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *arXiv preprint arXiv:2306.03310*, 2023.

Haoyu Liu, Sucheng Ren, Tingyu Zhu, Peng Wang, Cihang Xie, Alan Yuille, Zeyu Zheng, and Feng Wang. Spiral rope: Rotate your rotary positional embeddings in the 2d plane. *arXiv preprint arXiv:2602.03227*, 2026.

Quan Khanh Luu, Pokuang Zhou, Zhengtong Xu, Zhiyuan Zhang, Qiang Qiu, and Yu She. Manifeel: Benchmarking and understanding visuotactile manipulation policy learning. *arXiv preprint arXiv:2505.18472*, 2025.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. Leworld-model: Stable end-to-end joint-embedding predictive architecture from pixels. *arXiv preprint arXiv:2603.19312*, 2026.

Duc Huy Nguyen, Tim Schneider, Guillaume Duret, Alap Kshirsagar, Boris Belousov, and Jan Peters. Tacex: Gelsight tactile simulation in isaac sim–combining soft-body and visuotactile simulators. *arXiv preprint arXiv:2411.04776*, 2024.

NVIDIA. Isaac Sim. URL https://github.com/isaac-sim/IsaacSim.

OpenLoong Community. Baihu-vtouch: A cross-embodiment visuo-tactile multi-modal robot manipulation dataset. https://www.modelscope.cn/datasets/OpenLoong-Community/visuo-tactile, 2026. Accessed: 2026-09-25.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

Maram Sakr, Juyan Zhang, HF Machiel Van der Loos, Dana Kulić, and Elizabeth Croft. Consistency matters: Defining demonstration data quality metrics in robot learning from demonstration. *ACM Transactions on Human-Robot Interaction*, 15(2):1–31, 2025.

Lucy Xiaoyang Shi, Archit Sharma, Tony Z. Zhao, and Chelsea Finn. Waypoint-based imitation learning for robotic manipulation. In *Proceedings of The 7th Conference on Robot Learning*, pp. 2195–2209, 2023.

Modi Shi, Li Chen, Jin Chen, Yuxiang Lu, Chiming Liu, Guanghui Ren, Ping Luo, Di Huang, Maoqing Yao, and Hongyang Li. Is diversity all you need for scalable robotic manipulation? *IEEE Transactions on Robotics*, 2026.

Zilin Si and Wenzhen Yuan. Taxim: An example-based simulation model for gelsight tactile sensors. *IEEE Robotics and Automation Letters*, 7(2):2361–2368, 2022.

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, 2025. URL https://arxiv.org/abs/2508.10104.

Balakumar Sundaralingam, Siva Kumar Sastry Hari, Adam Fishman, Caelan Garrett, Karl Van Wyk, Valts Blukis, Alexander Millane, Helen Oleynikova, Ankur Handa, Fabio Ramos, Nathan Ratliff, and Dieter Fox. curobo: Parallelized collision-free minimum-jerk robot motion generation, 2023. URL https://arxiv.org/abs/2310.17274.

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*, 2026.

Shaoxiong Wang, Mike Lambeta, Po-Wei Chou, and Roberto Calandra. Tacto: A fast, flexible, and open-source simulator for high-resolution vision-based tactile sensors. *IEEE Robotics and Automation Letters*, 7(2):3930–3937, 2022.

Yansong Wu, Zongxie Chen, Fan Wu, Lingyun Chen, Liding Zhang, Zhenshan Bing, Abdalla Swikir, Sami Haddadin, and Alois Knoll. Tacdiffusion: Force-domain diffusion policy for precise tactile manipulation. In *2025 IEEE international conference on robotics and automation (ICRA)*, pp. 11831–11837. IEEE, 2025.

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.

Chao Xue, Chaofan Zhang, Wenxuan Ma, Guocai Yao, Shaowei Cui, and Shuo Wang. Hitac-wam: A hierarchical tactile world action model for contact-rich robot manipulation. *arXiv preprint arXiv:2608.19574*, 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. *arXiv preprint arXiv:2503.02881*, 2025.

Zhao-Heng Yin, Binghao Huang, Yuzhe Qin, Qifeng Chen, and Xiaolong Wang. Rotating without seeing: Towards in-hand dexterity through touch. *arXiv preprint arXiv:2303.10880*, 2023.

Kelin Yu, Yunhai Han, Qixian Wang, Vaibhav Saxena, Danfei Xu, and Ye Zhao. Mimictouch: Leveraging multi-modal human tactile demonstrations for contact-rich manipulation. *arXiv preprint arXiv:2310.16917*, 2023.

Ying Yuan, Haichuan Che, Yuzhe Qin, Binghao Huang, Zhao-Heng Yin, Kang-Won Lee, Yi Wu, Soo-Chul Lim, and Xiaolong Wang. Robot synesthesia: In-hand manipulation with visuotactile sensing. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6558–6565. IEEE, 2024.

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.

Xidong Zhang, Yichi Zhang, Jiaxin Shi, Fucai Zhu, Siyu Zhu, Michael Yu Wang, Xiaojun Wu, and Weihao Yuan. Unitacvla: Unified tactile understanding and prediction in vision language action models. *arXiv preprint arXiv:2606.31723*, 2026.

Jialiang Zhao, Yuxiang Ma, Lirui Wang, and Edward H Adelson. Transferable tactile transformers for representation learning across diverse sensors and tasks. *arXiv preprint arXiv:2406.13640*, 2024.

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

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.

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*, 2026.

A BENCHMARK DATA COLLECTION

A.1 AGENTIC MANIPULATION DATA GENERATION.

Motivation. Agentic data generation is particularly suitable for data collection in our tasks, especially for the `Precise` suite. We initially followed prior work Liu et al. (2023) and used a SpaceMouse to collect data. However, due to our strict success criteria and the limited precision of manipulation teleoperation, this approach proved difficult and inefficient, often resulting in low-quality action trajectories. With our carefully designed agentic data-generation pipeline, data collection becomes substantially more efficient, produces higher-quality trajectories, and scales more easily.

Settings. To collect scalable, high-quality manipulation data, we develop an agentic harness based on Codex to generate and debug agentic action experts on the Isaac Sim platform. We formulate agentic manipulation as a controller program with an explicit closed-loop structure, which makes decisions based on observations, robot states, and tactile information. The control process separates free-space motion from contact-rich or tactile-aware motion. Motion planning handles approaching and transporting objects in free space, while local closed-loop control executes grasping, alignment, insertion, pressing, fitting, and fastening. Tactile or geometric conditions, rather than fixed timing, trigger transitions between stages.

Environment & Sensors. We use IsaacSim 5.1 NVIDIA as the backend for realistic simulation and CuRobo Sundaralingam et al. (2023) for manipulation trajectory planning. We use two recently popular visual-tactile sensors, *i.e.*, GelSight Mini and DM-Tac. GelSight Mini and DM-Tac are vision-based optical tactile sensors that infer contact information from camera observations of elastomer deformation. GelSight Mini provides high-resolution sensing of contact geometry and texture, while DM-Tac emphasizes dense and dynamic tactile perception, including contact shape, pressure, tangential forces, and slip. The robot is a Franka arm with two grippers that are equipped with these two sensors. Because the two grippers have different geometries, we tuned the trajectories and data collection expert separately.

Success criteria. A task is considered successful only when the object reaches a stable seated state, such as full insertion, complete press fitting, or completion of the required amount of rotation. In discrimination and search tasks, decisions must be based on tactile observations collected through interaction. To ensure high-quality data, the success criteria used for data collection are stricter than those used for evaluation.

For agentic data collection, we find that trajectories judged successful during collection do not always succeed when replayed under the same scene configuration. This is because the collection process effectively performs closed-loop manipulation: rather than relying entirely on a predefined motion pattern, the agent dynamically replans and adjusts its actions in response to environmental changes. Meanwhile, the simulation environment includes some nondeterminism, and discrepancies can occur even when the same random seed is used. This issue is particularly pronounced for fine-grained manipulation tasks, where even small deviations can lead to task failure. In contrast, replay simply re-executes the recorded trajectory and cannot perform such dynamic corrections. Therefore, we consider only trajectories that also succeed during replay to be valid training trajectories.

Skills. Skills can extend the ability of the data collection agent, thus leading to a more accurate, efficient data collection pipeline. Given the pipeline stages and the need to construct a benchmark, we develop skills for simulation assets, task generation, environment building, and data generation, as shown in Table 7.

Built-in CLI. The agent needs to interact with the simulation dynamically and understand the physical problems during rollout. To improve the efficiency of data collection and help the agents generate trajectories and repair problems when generating the environment and rolling it out, we develop a set of built-in functions to let the agent read the relative simulation debug logs and the vision of the simulation, as well as automatically process the collected data, as shown in Table 8.

Human in the loop. We found that constructing a simulation environment, performing tactile-oriented tasks, and creating corresponding trajectory-generation experts solely through the agent are very difficult. Because of limitations in 3D spatial reasoning, physical information understanding, and rollout video understanding, agents often cannot identify failure reasons and automatically repair

Table 7: Summary of data collection skills.

Skill	Functional Scope
`expert-trajectory`	Development and refinement of expert policies for our benchmark, including tactile grasping, cuRobo-based free-space motion planning, insertion and placement, success-condition specification, and trajectory-smoothness optimization.
`dynamic-collect`	Collection and same-environment replay-gated evaluation of dynamic conveyor-belt tasks `tac_dynamic`, together with systematic diagnosis of Kit execution hangs and zero-episode dataset failures.
`reset-randomization`	Design and validation of reset-time perturbations for the benchmark suite, including `pose_range`, object permutation, and `layout_xy`, with consistency verification between JSON specifications and runtime perturbation bounds.
`generate-tactile-task`	Generation and validation of collectible tactile manipulation environments from natural-language task specifications.
`object-expert-repair`	Diagnosis and remediation of geometry and collision issues, insertion failures, false-positive success conditions, and expert trajectory collection consistent with teleoperated behaviour.

Table 8: Built-in Functions and Their Roles in the Data Processing Pipeline.

Function	Role in the Data Processing Pipeline
`collect`	Performs data collection, validation, and persistence while exposing diagnostic information and runtime task traces to support agent-driven analysis of collection failures and their underlying patterns.
`expert`	Dispatches the task-specific expert procedure to generate demonstrations for the target manipulation task.
`replay_gate`	Replays collect trajectories in the corresponding environment and evaluate execution correctness as a quality-control gate for dataset admission.
`capture`	Captures images from arbitrary tasks in Isaac Sim for visual inspection, debugging, and downstream multimodal processing.
`validate_dataset`	Inspects dataset metadata and verifies individual episodes against the predefined episode schema to ensure structural consistency.
`convert_dataset`	Converts datasets between formats, processes action sequences, and removes redundant or unnecessary frames to produce training-ready datasets.

problems, especially for tasks that require precise, contact-rich manipulation. To ensure the quality of simulation assets, environments, and training data, as well as fair evaluation, we use a human-in-the-loop approach across the entire data collection pipeline.

Touch in the collection. In addition to synchronously recording tactile RGB images, marker information, and other tactile signals during data collection, tactile sensing is also actively used throughout the agentic data collection process. During grasping or contact verification, we use the indentation depth measured by the tactile sensors as input to an adaptive gripper controller, which adjusts the gripper opening accordingly to ensure that the applied grasping force remains within the desired range. For sorting tasks, we leverage privileged information while using rule-based procedures to mimic the stochastic process by which humans identify objects through touch. For search tasks, during data collection, we determine whether the target object has been found based on whether the tactile sensor registers a contact response.

Randomness in data collection. During data collection, especially for the `Recognition` suite, we not only randomize the position in a small set of object positions, but we also change the object's position and randomize the order of the searching process to simulate the behavior, *e.g.*, a human uses the head to search and recognize the object he needs by touch.

Action Representation. We use 8D absolute joint actions with a continuous gripper. During data collection, we used EE for replay; the success rate is very low because IK errors reduce data-collection efficiency and data quality.

Force. We also provide force information for the two grippers. Each gripper pad records 6D force, *i.e.*, 3D XYZ force from each pad.

Other Modalities. Since our benchmark is built on IsaacSim, it can export depth, semantic masks, object tracking, object pose, and other information. We have also reserved the export interface.

A.2 Data Collection Experience & Discussion

Length of the trajectory. For demonstrations of the same task, trajectories should minimize redundant motion and unnecessary corrections. Excessive trajectory length increases the effective task horizon and may exacerbate compounding errors in imitation learning Ross et al. (2011); Shi et al. (2023); Zhao et al. (2023). Prior work also identifies path length and trajectory consistency as relevant indicators of demonstration quality Sakr et al. (2025). Therefore, operators should execute tasks efficiently while preserving motions necessary for safety and robustness.

Speed of the trajectory. Demonstrations should exhibit smooth and temporally consistent motion rather than strictly constant velocity. Abrupt acceleration, frequent stops, and oscillatory corrections introduce unnecessary action variability and may hinder policy learning. Motion smoothness, often characterized by jerk, is associated with natural, high-quality motion generation Flash & Hogan (1985); Sakr et al. (2025). Operators should therefore maintain stable motion in free space and smoothly decelerate during contact or precision-sensitive phases.

Consistency of the trajectories. Demonstrations of the same task should follow consistent motion strategies and execution patterns. Large inter-demonstration variations in trajectory geometry and motion characteristics can increase ambiguity in the learned action distribution. Prior work shows that consistency across demonstrations is strongly associated with both task success and generalization performance Sakr et al. (2025).

Predictability of the actions. Operators should avoid arbitrary variations in action selection under similar task states. Consistent state-action mappings reduce unnecessary action ambiguity and improve the predictability of demonstrations, which has been identified as an important indicator of robot data Shi et al. (2026).

A.3 TacMind-Bench Task List

The Standard suite comprises 28 tasks spanning the three evaluation regimes (`Precise`, `Recognition`, and `Dynamic`). All Standard tasks share the same observation interface (a third-person agent view and a wrist camera, two fingertip tactile sensors—one per gripper pad—and 8-D proprioception), instantiated with two gripper embodiments (GelSight Mini and DM-Tac), and each provides 100 expert demonstrations collected with the agentic pipeline described above. Figure 5 shows typical agent-view and tactile observations at a contact-rich moment, and the natural-language instruction of every task is listed in Table 9; both group the tasks by evaluation regime.

`Precise` regime (6 tasks). Tabletop contact regulation with static scenes: pick-and-insert of a tube, key, connector, or gear into a hole base, lock, or port; insertion of a power plug and a bulb.

`Recognition` regime (11 tasks). Pick-and-place in which the target is visually ambiguous or occluded and must be identified by touch (texture, ribs, edges, size): in seven tasks the objects are hidden in boxes or under covered trays, and in the remaining four they are visible but nearly identical.

`Dynamic` regime (11 tasks). Insertion and press-fitting under externally driven motion: either the target fixture (moving-target variants) or the source base (moving-source variants) is mounted on a translating conveyor platform, requiring precise, tactile-guided alignment under relative motion.

B Tac-LeWM Pretraining

Heterogeneous pretraining data. To improve the generality of Tac-LeWM beyond a single sensor and visual domain, we construct a heterogeneous visuo-tactile corpus from TacMind-Bench, ToucHD Feng et al. (2026), Daimon-Infinity Daimon Robotics (2026), RDP Xue et al. (2025), VLA-Touch Bi et al. (2026), and VisuoTactile-QINGLOONG OpenLoong Community (2026), comprising



Figure 5: The 28 standard tasks of TacMind-Bench. For every task, the agent view (left image of each entry) and the two GelSight Mini tactile views (top, left pad; bottom, right pad of the gripper) are captured at the same contact-rich moment of an expert demonstration, *i.e.*, the frame with maximum tactile deformation relative to the episode start. Entries are grouped by evaluation regime: `Precise` (6), `Recognition` (11), and `Dynamic` (11).

Table 9: All 28 TacMind-Bench Standard tasks, grouped by evaluation regime. For the `Dynamic` regime, the fixture is mounted on the moving conveyor platform unless stated otherwise.

Regime	Task	Language Instruction
`Precise`	01	Pick up the tube and insert it into the black rubber hole base.
	02	Pick up the key and insert it into the blue plastic lock.
	03	Pick up the connector and insert it into the orange port.
	04	Pick up the gear and insert it into the white painted base.
	05	Insert the three-prong power plug into the power socket.
	06	Insert the bulb into the bulb socket.
`Recog`	01	Pick the larger M16 nut instead of the M8 nut from the covered tray, judging size by touch.
	02	Retrieve the dimpled golf ball rather than the smooth ball from the covered bin using touch.
	03	Explore the covered compartment tray by touch, find the service token with the sawtooth edge, and place it on the rubber service mat.
	04	Among the caps in the covered recycling tray, find the cap with horizontal ribs—not vertical ribs—and place it in the recycle cup.
	05	Explore the two balls by touch, identify the dimpled golf ball rather than the smooth table-tennis ball, and place it in the left cup.
	06	Among the six equal-size cylinders in the open parts box, find the one with the most circumferential threads and place it on the inspection tray.
	07	Among the tokens in the covered compartment tray, find the one with the wave-shaped edge and place it on the rubber service mat.
	08	Among the three caps on the table, use touch to find the cap with horizontal ribs and place it in the bowl.
	09	Pick up the horizontal rectangular peg from the box bottom, lift it vertically, and insert it.
	10	Find and retrieve the target object from the box using tactile feedback.
	11	Distinguish the golf ball from the table tennis ball by touch and lift the golf ball.
`Dynamic`	01	Insert the three-prong power plug already held by the gripper into the power socket on the moving conveyor platform.
	02	Insert the bulb already held by the gripper into the bulb socket on the moving conveyor platform.
	03	Insert the USB plug already held by the gripper into the USB socket on the moving conveyor platform.
	04	Pick the test tube from the table and insert it into the angled hole base mounted on the conveyor platform.
	05	Pick the test tube from the moving source base and insert it into the angled hole base on the table.
	06	Pick the key from the table and insert it into the target lock mounted on the conveyor platform.
	07	Pick the key from the moving source lock and insert it into the target lock on the table.
	08	Pick the HDMI connector from the table and insert it into the HDMI slot mounted on the conveyor platform.
	09	Pick the HDMI connector and insert it into the HDMI slot on the table.
	10	Pick the gear from the table and press it into the gear base mounted on the conveyor platform.
	11	Pick the gear and press it into the gear base on the table.

approximately 3.0M frames and 42 hours of interaction. These datasets span different tactile sensors, camera configurations, robot platforms, and teleoperation interfaces, providing substantially broader contact and visual variation than TacMind-Bench alone. We map all observations to a common visual–tactile interface, with visual inputs resized to a shared spatial resolution and tactile observations represented using a consistent temporal layout. Actions require a different treatment because the datasets expose fundamentally different control parameterizations, including end-effector poses, wrist motions, joint-space commands, and gripper states. Rather than forcing these commands into a single kinematic representation, we retain their source-specific semantics and use a dedicated action embedding for each source before the shared dynamics predictor. Actions are additionally normalized within each source using statistics computed only from valid action samples. This prevents differences in physical scale or control parameterization from dominating the shared predictor.

Action-aware pretraining objective. A central challenge of heterogeneous pretraining is that visual and tactile context alone can often explain a large fraction of short-horizon state variation. Consequently, minimizing prediction error does not guarantee that the learned dynamics remain sensitive to the commanded action. This problem becomes more pronounced when the corpus contains samples without valid action supervision. We therefore separate representation learning from action-conditioned dynamics learning. Samples without valid actions may contribute to the representation objective, but they are excluded from the predictive and action-reliance terms rather than being interpreted as zero actions. For action-bearing samples, we explicitly encourage the predictor to distinguish the true action sequence from a shuffled alternative. The complete pretraining objective is

$$\mathcal{L} = \mathcal{L}_{\text{rep}} + \lambda_{\text{pred}} m^a \left\| \hat{z}^a_{t+1} - z_{t+1} \right\|_2^2 + \lambda_{\text{act}} m^a \left[\cos\left(\hat{z}^{\pi(a)}_{t+1}, z_{t+1} \right) - \cos\left(\hat{z}^a_{t+1}, z_{t+1} \right) + \delta \right]_+, \quad (5)$$

where $\mathcal{L}_{\text{rep}}$ denotes the visual–tactile representation objective, m^a indicates the availability of valid action supervision, and $\hat{z}^a_{t+1}$ and $\hat{z}^{\pi(a)}_{t+1}$ denote predictions conditioned on the true and shuffled actions, respectively. Actions are shuffled only among samples from the same source, preventing the predictor from satisfying the objective through dataset identification rather than action sensitivity. The target representation and shuffled conditioning branch are detached when evaluating the action-reliance term, so that this constraint primarily shapes the dynamics predictor instead of distorting the representation or action embedding spaces. We also randomly suppress visual observations during pretraining, encouraging the tactile pathway to retain predictive information rather than becoming redundant with vision.

Pretraining protocol. Although the unified corpus contains all six sources, the final training mixture emphasizes TacMind-Bench together with RDP, VLA-Touch, and VisuoTactile-QINGLOONG. TacMind-Bench receives the largest sampling mass because it defines the downstream task distribution, while the additional sources provide complementary contact dynamics and sensor diversity. ToucHD is not sampled in the final mixture because it provides no action supervision, and Daimon is excluded because its substantially larger volume would otherwise dominate the training distribution. This sampling strategy separates corpus diversity from source frequency: heterogeneous datasets broaden the representation space, while the optimization distribution remains anchored to action-bearing data relevant to downstream control. Combined with source-specific action embeddings, per-source normalization, masked action supervision, and the action-reliance objective, this formulation allows Tac-LeWM to exploit heterogeneous visuo-tactile data without sacrificing action-conditioned dynamics. The resulting pretrained model is used as the initialization for subsequent task-specific adaptation and control learning.

C VISUALIZATION

We visualize some additional cases on TacMind-Bench, as shown in Fig. 6 and Fig. 7.

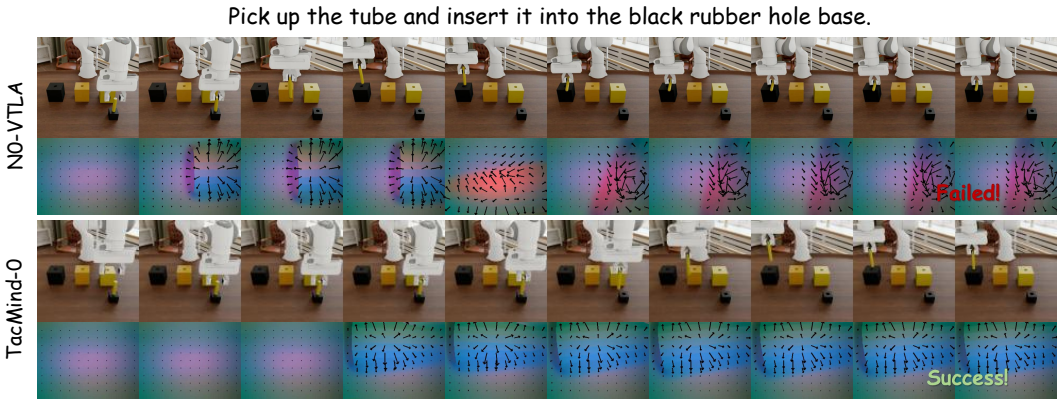

Figure 6: Comparision between N0-VTLA and TacMind-0. Task: Pick up the tube and insert it into the black rubber hole base.

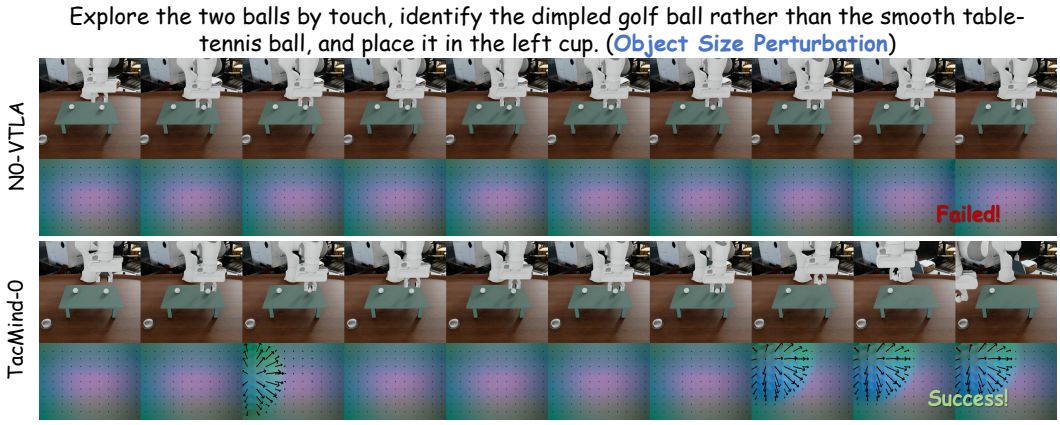

Figure 7: Comparision between N0-VTLA and TacMind-0. Task: Explore the two balls by touch, identify the dimpled golf ball rather than the smooth table-tennis ball, and place it in the left cup (Object Size Perturbation).

D Real-World Experiment

D.1 Experimental Setup

We conduct real-world experiments on a RealMan-ISF robotic arm equipped with a DM-Tac W tactile sensor to evaluate our method on fine manipulation and tactile discrimination tasks. Our benchmark comprises six subtasks across three categories: *Insert*, *Assemble*, and *Discriminate*. These categories evaluate precise alignment, contact-guided assembly, and object weight discrimination, respectively. The subtasks are defined as follows:

- **Insert–USB:** Align a USB connector with its corresponding port and insert it.
- **Insert–Plug:** Align a plug with its corresponding socket and insert it.
- **Assemble–Large Gear (GearL):** Align a large gear with the target assembly position and complete assembly through positional and rotational adjustments.
- **Assemble–Small Gear (GearS):** Align a small gear with the target assembly position and complete assembly through positional and rotational adjustments.
- **Discriminate–Heavy Bottle:** Identify the bottle containing an added weight through physical interaction.
- **Discriminate–Light Bottle:** Identify the light bottle without an added weight through physical interaction.



Figure 8: Representative trajectories for the six real-world tasks. From top to bottom, the rows show USB insertion, plug insertion, small-gear assembly, large-gear assembly, heavy-bottle discrimination, and light-bottle discrimination. Each row contains eight frames arranged in chronological order from left to right.

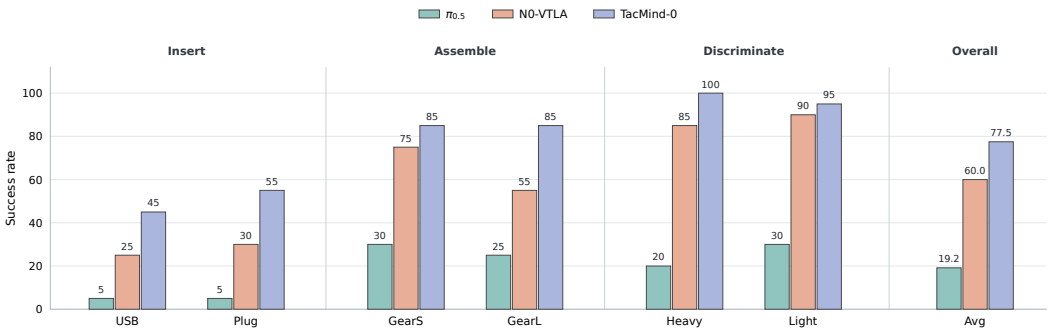

Figure 9: Success rates on six real-world manipulation tasks. We compare $\pi_{0.5}$, N0-VTLA, and TacMind-0 on two insertion tasks (USB and Plug), two assembly tasks (GearS and GearL), and two weight discrimination tasks (Heavy and Light). Each method is evaluated over 20 trials per task. The numbers above the bars denote success rates in percent, and Avg is the mean across the six tasks.

Representative trajectories for all six subtasks are shown in Fig. 8. Each row presents eight frames in chronological order, illustrating the progression from the initial scene to task completion.

The dataset contains 50 trajectories per subtask, yielding 300 trajectories in total. We use task success rate as the primary evaluation metric. Each method is evaluated over 20 trials per subtask, and success rate is computed as the fraction of successful trials.

D.2 Implementation Details

We compare our method with three baselines: $\pi_{0.5}$, N0-VTLA, and TacMind-0. All methods are trained for 40,000 steps using eight NVIDIA A100 GPUs with a global batch size of 64. For each method, we evaluate checkpoints at 20,000, 30,000, and 40,000 training steps and report the best-performing result among these checkpoints.

D.3 Task Analysis

Fig. 9 reports the success rates of the three methods across six real-world tasks. TacMind-0 achieves an average success rate of 77.5%, compared with 60.0% for N0-VTLA and 19.2% for $\pi_{0.5}$.

Insert. Insertion requires precise visual alignment and timely correction when contact reveals a positioning error. On USB and Plug insertion, $\pi_{0.5}$ succeeds in only 5% of trials for each task. Without tactile feedback, it has difficulty detecting an obstructed insertion and correcting its position. N0-VTLA improves the success rates to 25% and 30%, respectively, indicating that tactile feedback helps the policy recover from misalignment. TacMind-0 further raises the success rates to 45% on USB and 55% on Plug. These results suggest that its combination of manipulation precision and tactile feedback supports both initial alignment and corrective actions after contact.

Assemble. Gear assembly requires the policy to locate a visually occluded target position and respond to contact between the gears. After reaching the insertion position, the policy must detect whether the gears have meshed and adjust their relative rotation when necessary. $\pi_{0.5}$ achieves success rates of 30% on GearS and 25% on GearL, reflecting the difficulty of these operations without tactile feedback. N0-VTLA reaches 75% and 55%, respectively, but its lower performance on GearL indicates that tactile localization alone does not always lead to successful assembly. TacMind-0 achieves 85% on both tasks. Its consistent performance across the two gear sizes is consistent with more reliable tactile-guided positioning and contact-dependent rotational adjustment.

Discriminate. Bottle discrimination requires the policy to determine whether a bottle contains an added weight through physical interaction. $\pi_{0.5}$ succeeds in 20% of Heavy trials and 30% of Light trials. Both rates are below the 50% chance level for a binary choice, as task success also requires the correct choice to be executed successfully. With tactile feedback, N0-VTLA reaches 85% on Heavy and 90% on Light. TacMind-0 obtains the highest success rates, achieving 100% and 95%, respectively. These results show that TacMind-0 can reliably use interaction feedback to distinguish the bottles and complete the corresponding actions.