ME-DEX 1.0: UNIFYING HETEROGENEOUS TACTILE SENSING INTO WORLD ACTION MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

World Action Models predict future visual states to inform robot action generation. Tactile sensing directly captures contact states and force changes during physical interaction. Existing methods often treat tactile sensing as a conditioning input for action generation, thereby modeling only the current contact state. Instead, tactile signals should be modeled alongside video as future observations of the evolving world state. We present ME-Dex 1.0 (ME-Dex), a unified World Action Tactile Model for joint modeling of visual states, tactile states, and robot actions. ME-Dex adopts a Mixture-of-Transformers comprising a Video Expert, a Tactile Expert, and an Action Expert, all trained with flow matching. Multimodal joint attention connects the three experts at intermediate layers, allowing action generation to exploit learned visual and tactile dynamics during joint generation. To support heterogeneous tactile inputs across embodiments, ME-Dex maps sensor surfaces to a Canonical Hand Model and encodes the aligned observations into a unified latent space with a Unified Tactile Autoencoder. We augment tactile-free simulation trajectories with synchronized tactile observations for joint visual, tactile, and action learning. Experiments on RoboTwin, DexJoCo, and ManiFeel improve average manipulation success over the strongest baselines by 11.2, 8.9, and 15.0 percentage points. Qualitative rollouts on physical robots further demonstrate the use of visual and tactile observations across grippers and dexterous hands. Code and supplementary materials are available at `https://anonymousmedex.github.io/`.

1 INTRODUCTION

Embodied AI aims to build agents that perceive, reason, and act through interaction with the physical world. Vision-Language-Action models (VLAs) connect visual observations and language instructions to executable robot actions, drawing on knowledge from pretrained vision-language models (Kim et al., 2025; Black et al., 2025b; Wang et al., 2026c;d). World Action Models (WAMs) jointly model robot actions and future visual states, bringing the dynamics learned by video generation models into action prediction (Bi et al., 2026a; Li et al., 2026a; Ye et al., 2026; Ma et al., 2026).

Vision provides a rich physical interaction foundation by describing the scene and how objects move, while physical interaction also depends on how a robot applies force to an object and how the object responds. Tactile measurements complement visual observations by directly capturing this physical contact. For example, tactile signals changes during a grasp can help a robot assess whether an object is held securely. Modeling visual and tactile dynamics together therefore offers a path toward a more complete account of the world in which robots act.

Recent works (Zhao et al., 2025; Heng et al., 2025; Zhu et al., 2026) have incorporated tactile sensing into robot policies. As illustrated in Figure 1(a–b), a common approach encodes tactile observations as conditioning inputs, while recent methods have begun to predict future tactile states explicitly (Zheng et al., 2026b; Tian et al., 2026). Extending predictive tactile modeling across robots requires scalable tactile data and a shared model that can learn across embodiments.

However, a unified model must overcome three gaps. The Data Gap stems from lacking native tactile measurements in datasets like RoboTwin (Chen et al., 2026) and DexJoCo (Wang et al., 2026a), while existing tactile datasets like HATO (Lin et al., 2025) and ManiFeel (Luu et al., 2025) remain limited in scale and heterogeneous in format. Synchronized collection of paired tactile and

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.

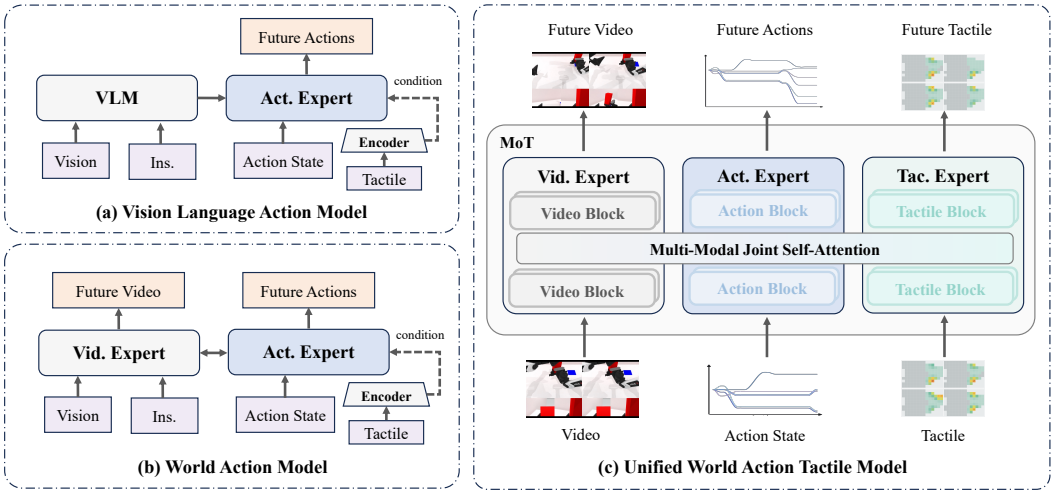

Figure 1: **From tactile conditioning to unified world action tactile modeling.** (a) Vision-language-action models generate actions from visual observations and task instructions, optionally conditioned on current tactile observations. (b) World action models jointly predict future video and actions, optionally conditioned on current tactile observations. (c) ME-Dex jointly predicts future video, tactile states, and actions within a unified model.

action data is also costly. The Representation Gap stems from substantial differences in tactile inputs across sensors and embodiments, making them difficult to joint training without a common spatial and latent representation. The Modeling Gap concerns how to jointly model future visual observations, tactile states, and robot actions within a predictive architecture, while making future tactile dynamics accessible to action generation.

To address these gaps, we introduce ME-Dex 1.0 (ME-Dex). We augment tactile-free RoboTwin and DexJoCo trajectories with tactile signals recorded from simulated force sensors during replay. The resulting observations are aligned with the corresponding visual observations and actions. We then map tactile measurements from dexterous hands and grippers to a Canonical Hand Model inspired by the human hand, and use a Unified Tactile Autoencoder to encode the aligned measurements into a shared latent space. At the model level, we place tactile states on equal footing with video when modeling the future world state. As shown in Figure 1(c), ME-Dex adopts a three-expert Mixture-of-Transformers (MoT) architecture. The Video Expert predicts future video, while the Tactile Expert predicts future tactile latents that capture the evolution of physical interactions. Multi-modal joint self-attention connects the three experts at intermediate layers, allowing the Action Expert to exploit future visual and tactile dynamics during joint generation. The three experts are jointly optimized for future video prediction, future tactile prediction, and action generation.

Across simulated benchmarks, ME-Dex achieves 82.5% average success rates on RoboTwin under Clean-to-Random protocol, 65.3% on DexJoCo, and 70.0% on ManiFeel, outperforming the strongest baselines by 11.2, 8.9, and 15.0 percentage points, respectively. We further perform qualitative evaluations on a LeRobot SO-101 equipped with a Paxini PX6AX gripper and on Xynova Flex2 dexterous hands, demonstrating manipulation from visual and tactile observations. Overall, our contributions can be summarized as follows:

- We present ME-Dex, a unified World Action Tactile Model built on a three-expert Mixture-of-Transformers architecture. The Video, Tactile, and Action Experts are jointly optimized to predict future video, future tactile latents, and action chunks, allowing action generation to exploit both visual and tactile dynamics.
- We propose a Canonical Hand Model inspired by human hand and design a Unified Tactile AE to unify heterogeneous tactile. These components align diverse dexterous hands and grippers to a common template and encode their tactile data into a shared latent space.
- We augment tactile-free RoboTwin and DexJoCo with aligned tactile data recorded from simulated force sensors, providing paired data for joint visual, tactile, and action learning across embodiments.

2 Related Work

2.1 Vision-Language-Action Models and World Action Models.

VLAs leverage pretrained vision-language knowledge to translate visual inputs and language instructions into executable robot actions. The π_0 (Black et al., 2025b) family combines a pretrained vision-language backbone with a flow-based action expert, and $\pi_{0.5}$ (Black et al., 2025a) extends this approach toward open-world generalization. X-VLA (Zheng et al., 2026a) explores learning across embodiments through soft prompts. These methods learn action generation from multimodal observations, with future environment states typically represented implicitly. World Action Models make future visual prediction an explicit part of action learning. Motus (Bi et al., 2026a) and MotuBrain (Motubrain Team et al., 2026) combine video generation, action generation, and semantic understanding within a MoT architecture. Fast-WAM (Yuan et al., 2026) studies how future video supervision supports action learning when video generation is omitted during inference. These approaches bring the predictive capabilities of video models into robot control. Our ME-Dex builds on this foundation by modeling future tactile states alongside future visual observations.

2.2 Tactile-Conditioned and Predictive Robot Policies.

Tactile observations complement visual perception by providing direct information about physical interaction. Some tactile-conditioned policies use tactile signals as condition input. Tactile-VLA (Huang et al., 2025) integrates tactile signals into VLA learning for contact-aware manipulation. FARM (Helmut et al., 2026) conditions a diffusion policy on tactile features and jointly predicts robot pose, gripper width, and gripping force. DECO (Li et al., 2026b) introduces a tactile adapter through cross-attention into a multimodal diffusion transformer. VLA-Touch (Bi et al., 2026c) combines tactile feedback for high-level planning with a diffusion controller for action refinement. T-Rex (Niu et al., 2026) adopts a variable-rate MoT architecture to combine slower visuomotor processing with faster tactile feedback. Motus2 (Bi et al., 2026b) uses a lightweight tactile expert for action refinement and future force prediction. These methods generally treat tactile signals as current conditions or task-specific prediction targets, without jointly modeling heterogeneous tactile inputs, future visual and tactile states, and actions across embodiments.

2.3 Heterogeneous Tactile Representation.

Learning from heterogeneous tactile data requires representations that accommodate different sensors and sensing layouts. UniTouch (Yang et al., 2024) aligns tactile representations with semantic information from other modalities. AnyTouch (Feng et al., 2025) studies tactile representation learning across sensors and downstream tasks. Sparsh-X (Higuera et al., 2025) extends self-supervised tactile learning to multimodal signals from the DIGIT360 sensor. These methods explore shared representations across sensor types and complementary sensing modalities, while do not establish a common spatial reference across robot embodiments or model world evolution by predicting future tactile. ME-Dex addresses these limitations with a Canonical Hand Model for spatial alignment and a Unified Tactile Autoencoder for mapping aligned observations into a shared latent space, enabling grippers and dexterous hands to participate in joint visual, tactile, and action modeling.

3 Method

3.1 Overview

As shown in Figure 2, ME-Dex jointly predicts future video, tactile states, and action chunks, treating visual and tactile signals as complementary observations of the future world state. Video captures scene evolution, while tactile sensing captures changes in physical contact.

The framework comprises two components. The Unified Tactile Representation maps heterogeneous tactile measurements across embodiments onto the Canonical Hand Model and encodes them into a unified latent space while preserving local force structure. The World Action Tactile Model uses a Mixture-of-Transformers with Video, Tactile, and Action Experts connected through multi-modal joint self-attention at intermediate layers. These connections enable the Action Expert to interact with representations of future visual and tactile dynamics during joint generation.

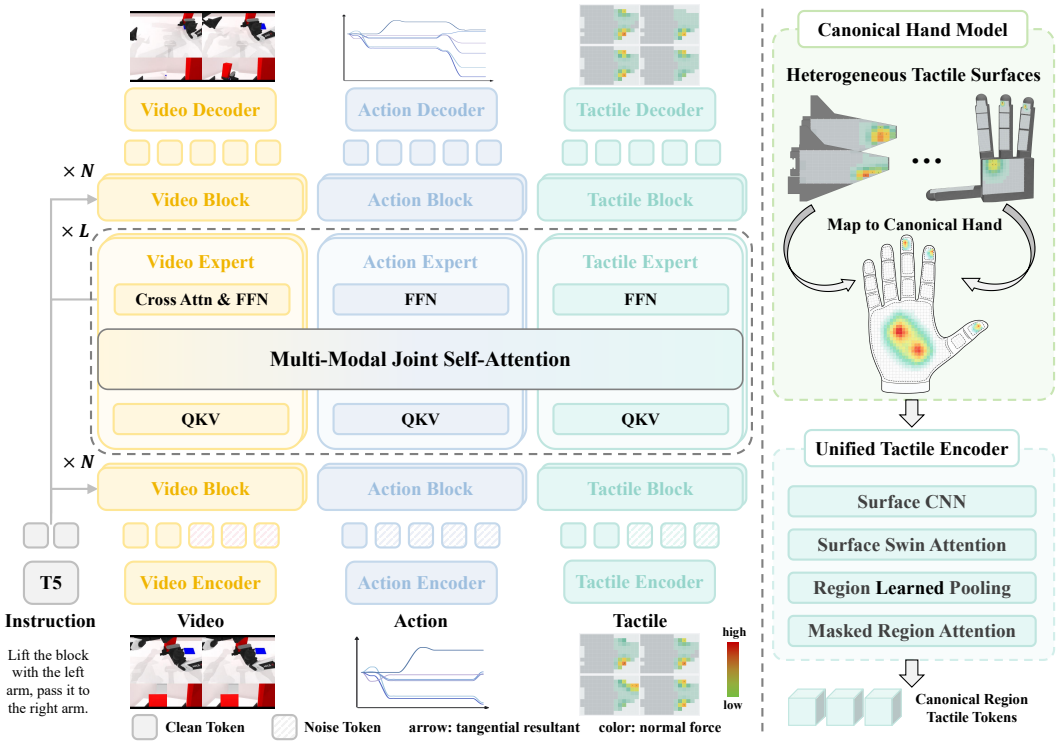

Figure 2: **Overview of ME-Dex.** The World Action Tactile Model (left) jointly predicts future video, tactile states, and action chunks using Video, Tactile, and Action Experts trained with conditional flow matching. Multi-modal joint self-attention connects the experts at intermediate layers. The Unified Tactile Representation (right) maps heterogeneous tactile measurements to canonical hand regions and encodes them into canonical region tokens.

3.2 Problem Formulation

World Action Models jointly predict future visual observations and robot actions, allowing visual prediction to inform action generation. ME-Dex extends this framework by modeling tactile states alongside video as future observations of the evolving world.

At time t, the model receives a task instruction ℓ, a visual observation $\mathbf{I}_t$ serving as the first video frame, a proprioceptive state $\mathbf{s}_t$, and the current tactile observation $\mathbf{T}_t$. Visual and tactile encoders map the observations into their respective latent spaces, giving the conditioning variables

$$\mathbf{z}_t^{\text{vis}} = E_{\text{vis}}(\mathbf{I}_t), \qquad \mathbf{z}_t^{\text{tac}} = E_{\text{tac}}(\mathbf{T}_t), \qquad \mathcal{C}_t = (\ell, \mathbf{s}_t, \mathbf{z}_t^{\text{vis}}, \mathbf{z}_t^{\text{tac}}). \tag{1}$$

The tactile observation contains the available signals from both hands, organized according to the canonical representation defined in Section 3.3.

Let $\mathbf{A}^+ = (\mathbf{a}_t, \ldots, \mathbf{a}_{t+k-1})$ denote a chunk of k actions, and let $\mathbf{V}^+$ and $\mathbf{T}^+$ denote future video and tactile sequences covering the same physical time window. The superscript $+$ denotes future sequences. Their latent representations are

$$\mathbf{Z}^{\text{vis},+} = E_{\text{vis}}(\mathbf{V}^+), \qquad \mathbf{Z}^{\text{tac},+} = E_{\text{tac}}(\mathbf{T}^+), \qquad \mathbf{Z}^{\text{world},+} = \left(\mathbf{Z}^{\text{vis},+}, \mathbf{Z}^{\text{tac},+}\right), \tag{2}$$

where E_{tac} operates frame by frame on a tactile sequence. ME-Dex jointly models these future observations and actions:

$$p_\theta(\mathbf{A}^+, \mathbf{Z}^{\text{world},+} \mid \mathcal{C}_t) = p_\theta\left(\mathbf{A}^+, \mathbf{Z}^{\text{vis},+}, \mathbf{Z}^{\text{tac},+} \mid \mathcal{C}_t\right). \tag{3}$$

During inference, the model jointly samples the action chunk and future observation latents, retaining the action chunk as the policy output. Ground-truth future video and tactile states are used as training targets and are unavailable at inference.

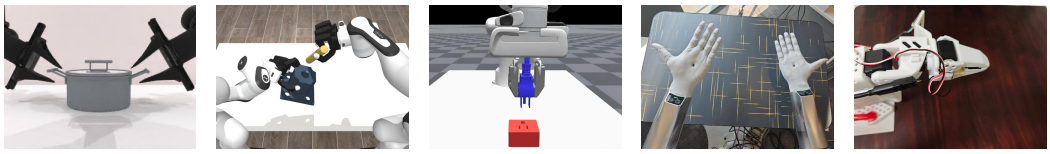

Figure 3: **Multi-source heterogeneous tactile inputs.** From left to right: RoboTwin, DexJoCo, ManiFeel, Xynova Flex2 dexterous hands, and a tactile-equipped LeRobot SO-101 gripper.

3.3 Unified Tactile Representation

Our tactile inputs come from diverse grippers and dexterous hands across simulation and real-world platforms, as shown in Figure 3. Tactile sensors across robot embodiments differ in surface geometry, spatial resolution, sensing coverage, and data format. We represent each sensing surface r as a three-channel force field $\mathbf{F}_t^{(r)} \in \mathbb{R}^{3\times H\times W}$, containing one normal and two tangential force components on an $H \times W$ grid. A layout mask identifies valid sensing positions, including positions without contact.

3.3.1 Canonical Hand Model

To give tactile measurements a consistent spatial meaning across embodiments, we design a Canonical Hand Model inspired by the human hand and treat grippers and dexterous hands as morphological variations of this template. The template defines corresponding finger and palm regions for the left and right hands. Region assignments establish a common spatial reference without requiring identical surface geometry across end effectors.

Figure 2 illustrates the mapping. Each dexterous-hand sensing surface is assigned to its corresponding finger or palm region. The two sensing surfaces of each RoboTwin gripper map to the thumb and index regions, respectively. Different sources observe different subsets of the template. A region mask distinguishes unavailable regions from observed regions without contact, preserving the difference between missing measurements and measured absence of contact.

3.3.2 Unified Tactile Autoencoder

Canonical mapping aligns sensing locations, but force fields still differ in resolution, geometry, and coverage. We therefore use a shared Unified Tactile Autoencoder to encode each tactile frame into compact canonical region tokens while preserving local contact structure.

The encoder extracts local force features, aggregates valid positions through masked pooling, and exchanges information among tokens in a fixed canonical order using masked Transformer layers. Unobserved regions remain masked with zero outputs. The decoder maps the tokens back to their sensing surfaces and predicts force components and contact probabilities, which are combined with the valid-position mask to reconstruct the force fields.

We train the autoencoder with force reconstruction, contact supervision, temporal change matching, and latent regularization. After pretraining, the autoencoder is frozen and provides current tactile conditions, future tactile targets, and the decoder for predicted tactile latents. Details of the preprocessing, architecture, and training objective are provided in Appendix A.

3.4 World Action Tactile Model

3.4.1 Three-Expert Architecture

The World Action Tactile Model uses a Mixture-of-Transformers architecture with three modality-specific Transformers: a Video Expert, a Tactile Expert, and an Action Expert. The experts are jointly trained with conditional flow matching on future video, tactile, and action targets. Each expert retains its own representation and hidden width, while their common depth supports interaction at intermediate layers.

During training, the Video Expert receives the encoded current video frame and noisy future video latents, the Tactile Expert receives the current tactile representation and noisy future tactile latents, and the Action Expert receives the proprioceptive state and noisy ground-truth action chunks.

The Tactile Expert treats tactile sensing as a temporal observation modality analogous to video. It predicts a spatially organized latent sequence for the future tactile frames and exchanges intermediate features with the Video and Action Experts. The frozen tactile decoder maps each predicted latent frame back to a force field, allowing contact dynamics to participate in world action modeling alongside visual dynamics.

3.4.2 Multi-Modal Joint Self-Attention

Inspired by H-Bridge (Wang et al., 2026b), we restrict cross-modal interaction to an intermediate set of Transformer layers. At each bridge layer, the three experts project their hidden states into a common attention space, apply multi-modal joint self-attention over video, tactile, and action tokens, and map the outputs back to their modality-specific widths. Outside these layers, each expert processes its tokens independently. This design preserves modality-specific processing while allowing the Action Expert to access representations of future visual evolution and contact dynamics during joint generation.

4 Experiments

4.1 Experimental Setup

Datasets and Platforms. We evaluate ME-Dex on three simulated benchmarks: RoboTwin provides 27,500 trajectories across 50 tasks for mixed Clean/Random training and evaluation; a separate Clean-only setting uses 2,500 trajectories to test Clean-to-Random generalization (Chen et al., 2026); DexJoCo provides 1,100 demonstrations across 11 tasks (Wang et al., 2026a); and ManiFeel, with 50 demonstrations per task across 4 insertion tasks and a separate policy for each task (Luu et al., 2025). We additionally conduct qualitative evaluations on a LeRobot SO-101 with a Paxini sensor and a dual-arm system equipped with Xynova Flex2 dexterous hands.

Baselines. On RoboTwin, we compare with $\pi_{0.5}$, Fast-WAM, and Motus. On DexJoCo, we compare with DP-T (Chi et al., 2025), $\pi_{0.5}$, DECO, and DECO.p, where DECO.p denotes the official tactile-adapted DECO variant evaluated on our tactile-augmented data. On ManiFeel, we use the official policy as the baseline. To study the effect of current tactile observations, we construct tactile-conditioned variants $\pi_{0.5}$+TAC, Fast-WAM+TAC, and ME-Dex(Tac Cond.). These variants encode tactile inputs with the pretrained Unified Tactile Autoencoder and append the resulting tokens to the policy conditioning context; all remaining settings are unchanged. More details of these variants are described in the Appendix B.

Table 1: **Multi-task manipulation performance on RoboTwin.** Success rates (%) under Clean + Random training. Train and Eval indicate current tactile observation usage. Zeroed denotes zero tactile input at evaluation. VA models video and action prediction with joint attention at all layers. Tac Cond. adds current tactile conditions, and Full Joint additionally predicts future tactile states.

Method	Tactile obs.		Future tactile prediction	H-Bridge	Success rate ↑	
	Train	Eval			Clean	Random
Baselines						
$\pi_{0.5}$	No	Not used	No	—	52.76	60.74
Fast-WAM	No	Not used	No	—	89.22	89.22
Motus	No	Not used	No	—	88.66	87.02
$\pi_{0.5}$+TAC	Yes	Observed	No	—	52.60	57.66
Fast-WAM+TAC	Yes	Observed	No	—	90.22	90.22
ME-Dex and variants						
ME-Dex (VA)	No	Not used	No	No	88.74	86.90
ME-Dex (Tac Cond.)	Yes	Observed	No	No	89.04	89.54
ME-Dex (Full Joint)	Yes	Observed	Yes	No	89.36	90.60
ME-Dex	Yes	Observed	Yes	Yes	91.56	**91.92**
ME-Dex (Zero Tac Input)	Yes	Zeroed	Yes	Yes	**91.70**	91.74

Table 2: **Multi-task manipulation performance on RoboTwin under Clean training.** Results are success rates (%). Baseline results are taken from the RoboTwin leaderboard (Chen et al., 2026). Tactile observations are set to zero for ME-Dex under both evaluation configurations.

Method	Clean→Clean	Clean→Random	Average
Fast-WAM	77.8	1.9	39.9
$\pi_{0.5}$	70.7	46.0	58.4
4D-WAM	81.5	41.8	61.6
GigaBrain-0.7	66.8	67.9	67.3
OLA-Sem	75.1	67.6	71.3
ME-Dex	**88.7**	**76.2**	**82.5**

Table 3: **Multi-task manipulation performance on DexJoCo.** Success rates (%) under the *rand-obj* configuration. DP-T, $\pi_{0.5}$, and DECO use no tactile input, whereas $\pi_{0.5}$+TAC, DECO.p, and ME-Dex use tactile data. DECO.p is the tactile-adapter variant of DECO. The suffix /B denotes a bimanual task, and Bimanual avg. is the equal-weight mean across the five bimanual tasks.

Task	DP-T	$\pi_{0.5}$	$\pi_{0.5}$+TAC	DECO	DECO.p(+TAC)	**ME-Dex**
Hammer Nail	58.7	40.0	10.0	68.0	76.0	**86.0**
Click Mouse	38.7	76.0	80.0	**90.0**	80.0	76.0
Pick Bucket	55.3	82.0	46.0	80.0	84.0	**92.0**
Pinch Tongs	6.0	16.0	0.0	46.0	60.0	**92.0**
Fold Glasses	11.3	24.0	40.0	50.0	**62.0**	56.0
Water Plant	60.0	66.0	82.0	**88.0**	76.0	72.0
Unlock iPad/B	0.0	8.0	0.0	40.0	32.0	**48.0**
Hanoi/B	8.0	14.0	2.0	18.0	24.0	**32.0**
Assembly/B	1.3	4.0	6.0	0.0	6.0	**28.0**
Microwave/B	42.7	70.0	44.0	64.0	68.0	**88.0**
Photograph/B	28.0	4.0	22.0	**76.0**	68.0	48.0
Bimanual avg.	16.0	20.0	14.8	39.6	39.6	**48.8**
Avg.	33.2	36.7	30.2	56.4	57.8	**65.3**

4.2 QUANTITATIVE COMPARISON

4.2.1 ROBOTWIN BENCHMARK

Mixed Clean and Random Training. Under mixed Clean and Random training, as shown in Table 1, ME-Dex achieves success rates of 91.56% and 91.92% under the Clean and Random configurations, respectively. These results exceed Fast-WAM by 2.34 and 2.70 percentage points, and Fast-WAM+TAC by 1.34 and 1.70 percentage points. Within the ME-Dex variants, adding current tactile conditions improves Random success from 86.90% to 89.54%. Jointly predicting future tactile states further increases it to 90.60%.

The effect of tactile conditioning varies across external baselines: Fast-WAM gains 1.00 percentage point, whereas $\pi_{0.5}$ decreases by 3.08 percentage points. When the current tactile input is zeroed, ME-Dex achieves 91.70% on Clean and 91.74% on Random, remaining close to the performance with observed tactile input.

Training on Clean Data Only. We further evaluate generalization from Clean to Random configurations by training ME-Dex on Clean-only Data. As shown in Table 2, ME-Dex achieves 88.7% under Clean→Clean evaluation and 76.2% under Clean→Random evaluation, with an average success rate of 82.5%. This exceeds the strongest listed baseline, OLA-Sem, by 11.2 percentage points on average. **As of September 26, the reporting results have been updated on the Robotwin official leaderboard and ranked first** (Chen et al., 2026).

4.2.2 DEXJOCO BENCHMARK

Under multi-task training across all 11 tasks in the *rand-obj* configuration, ME-Dex achieves the highest average success rate of 65.3%, as shown in Table 3. It outperforms DECO by 8.9 percentage points and DECO.p(+TAC) by 7.5 percentage points, and ranks first on seven of the 11 tasks.

Table 4: **Manipulation performance on ManiFeel.** Success rates (%) with single-task training and 50 evaluation episodes per task.

Method	Peg Insertion	USB Insertion	Power Plug Insertion	Gear Assembly	Average ↑
ManiFeel	40.0	56.0	58.0	**66.0**	55.0
ME-Dex	**58.0**	**74.0**	**88.0**	60.0	**70.0**

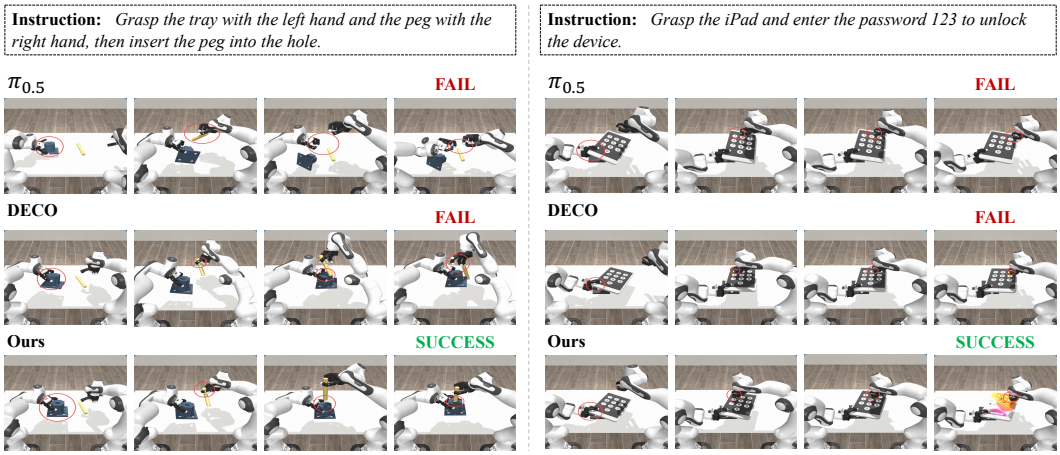

Figure 4: **Qualitative comparisons on DexJoCo.** Rollouts of $\pi_{0.5}$, DECO, and ME-Dex on Assembly (left) and Unlock iPad (right). Frames progress from left to right.

On the five bimanual tasks, ME-Dex achieves a bimanual average success rate of 48.8%, compared with 39.6% for both DECO and DECO.p. It ranks first on four of the five bimanual tasks, with particularly strong performance on Assembly and Microwave, while DECO and DECO.p remain stronger on Photograph.

Among the single-arm tasks, ME-Dex achieves the highest success rate on Pick Bucket and Pinch Tongs, reaching 92.0% on both. Its performance is not uniform across tasks: DECO.p performs better on Fold Glasses, while DECO leads on Click Mouse and Water Plant.

4.2.3 ManiFeel Benchmark

The ManiFeel baseline uses Diffusion Policy conditioned on visual observations and three-dimensional tactile force fields. ME-Dex jointly predicts future video, tactile states, and action chunks, allowing the experts to exchange intermediate representations during prediction.

Table 4 shows that ME-Dex outperforms this baseline on three of four tasks, achieving an average success rate of 70.0%, a gain of 15.0 percentage points. The largest improvement occurs on Power Plug Insertion, where success increases from 58.0% to 88.0%.

4.3 Qualitative Results

4.3.1 DexJoco

The Assembly and Unlock iPad examples illustrate the benefits of ME-Dex on contact-rich bimanual tasks. Both tasks require one hand to stabilize an object while the other performs precise interactions, with tactile observations available from both hands. As shown in Figure 4, ME-Dex coordinates peg insertion and password entry more effectively than $\pi_{0.5}$ and DECO.

4.3.2 RoboTwin

As shown in Figure 5, ME-Dex successfully completes both object scanning and drawer placement, coordinating object grasping with subsequent manipulation. In contrast, $\pi_{0.5}$ and Fast-WAM fail in both examples, while Fast-WAM+Tac completes drawer placement but fails the scanning task. These rollouts illustrate the advantage of ME-Dex on manipulation sequences requiring coordinated object interaction.

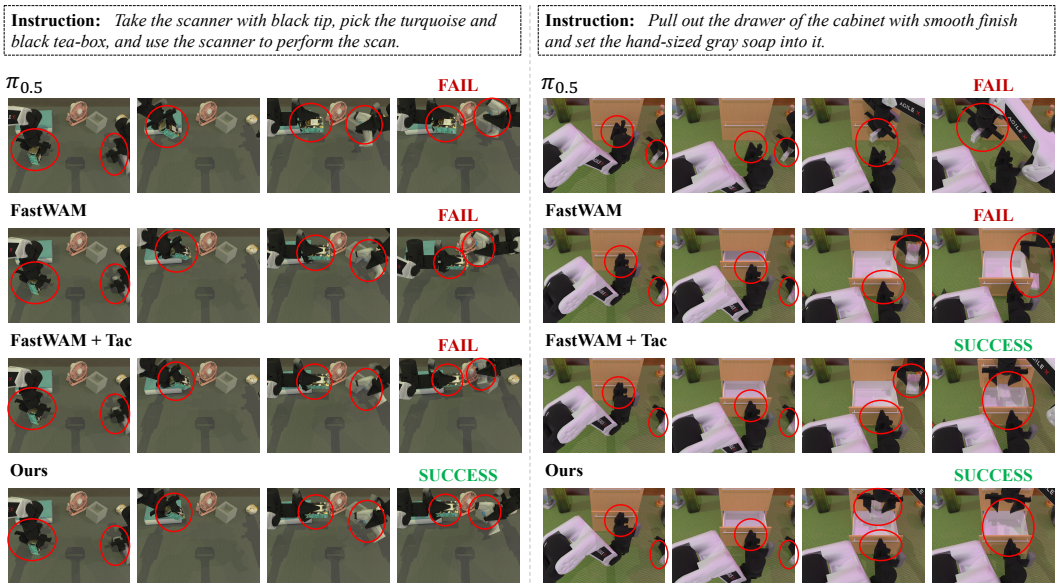

Figure 5: **Qualitative comparisons on RoboTwin.** Rollouts of $\pi_{0.5}$, Fast-WAM, Fast-WAM+TAC, and ME-Dex on object scanning (left) and drawer placement (right).

Table 5: **Tactile reconstruction across platforms.** Metrics are averaged equally across subsets within each platform. F1 scores are in percent and MAEs in source-specific normalized units.

Platform	Uniformly sampled windows		High-change windows	
	Contact F1 ↑	Contact force MAE ↓	Transition F1 ↑	Transition delta MAE ↓
RoboTwin	96.56	0.01797	89.73	0.01849
DexJoCo	90.05	0.02807	68.73	0.04293
ManiFeel	99.69	0.05478	66.96	0.06899

4.4 UNIFIED TACTILE REPRESENTATION

We evaluate a shared Unified Tactile Autoencoder on tactile data from RoboTwin, DexJoCo, and ManiFeel. We set the training and test sets with 8:2 ratio, both training subsets are combined for joint training. We evaluate both uniformly sampled and high-change windows. The former measure contact F1 and force MAE at contact locations, while the latter measure contact-transition F1 and frame-to-frame force MAE around contact onset and release. Metrics are averaged equally across subsets in each platform. As shown in Table 5, the shared autoencoder achieves contact F1 scores of 96.56%, 90.05%, and 99.69% on RoboTwin, DexJoCo, and ManiFeel, respectively. It also achieves transition F1 scores of 89.73%, 68.73%, and 66.96%, indicating that the representation retains information about both contact states and their temporal changes. Force MAEs are reported in source-specific normalized spaces and should therefore be interpreted separately for each platform.

5 CONCLUSION

We presented ME-Dex, a unified World Action Tactile Model that models tactile states alongside video as future observations for action generation. Its three-expert Mixture-of-Transformers jointly predicts future video, future tactile latents, and action chunks with flow matching, allowing action generation to exploit both visual and contact dynamics. A Canonical Hand Model and a Unified Tactile Autoencoder align heterogeneous tactile observations across embodiments and encode them into a shared latent space. We supplement tactile-free simulation trajectories with synchronized tactile observations to support joint visual, tactile, and action learning. Experiments on RoboTwin, DexJoCo, and ManiFeel demonstrate improved manipulation performance across grippers and dexterous hands, while qualitative rollouts show deployment on physical robot platforms.

Ethics Statement

This work uses publicly available robotic manipulation datasets, including RoboTwin, DexJoCo, and ManiFeel, together with data collected from controlled physical-robot experiments. No human subjects, animal experiments, or personally identifiable information are involved. The public datasets were used in accordance with their respective licenses and usage guidelines. The physical experiments were conducted in controlled tabletop settings. We identified no additional ethical concerns specific to this work.

Reproducibility Statement

We describe the model architecture, tactile preprocessing, canonical mapping, data construction, training and inference procedures, and evaluation protocols in the main paper and Appendices A–C. The appendices also provide task instructions, quantitative results, and additional qualitative rollouts for the physical-robot experiments. The benchmark datasets used in this work are publicly available and properly cited. Code, configuration files, and supplementary videos are provided in the anonymous supplementary material and at `https://anonymousmedex.github.io/`.

References

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, Hongyan Zhao, Hanyu Liu, Zhizhong Su, Lei Ma, Hang Su, and Jun Zhu. Motus: A Unified Latent Action World Model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 35101–35113, 2026a.

Hongzhe Bi, Zihao Zhou, Yihang Tang, Jingrui Pang, Shuhe Huang, Haitian Liu, Runqing Wang, Shuai Huang, Yichen Wang, Yiming Cheng, Ruowen Zhao, Zhenghua Li, Hengkai Tan, Xiaolong Liu, Jinhui Wan, Jiabao Liu, Min Zhao, Fan Bao, and Jun Zhu. Motus2: A Self-Evolving General World Model for Dexterous Manipulation. arXiv preprint arXiv:2608.30237, 2026b.

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, 2026c.

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, S Nair, Quan Vuong, Waluraj, Karl Pertsch, Lucy Xiaoyang Shi, Laura Smith, Anna Tanner, Jamesling, 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.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, Weiliang Deng, Yubin Guo, Tian Nian, Xuanbing Xie, Qiangyu Chen, Kailun Su, Tianling Xu, Guodong Liu, Mengkang Hu, Huan-ang Gao, Kaixuan Wang, Zhixuan Liang, Yusen Qin, Xiaokang Yang, Ping Luo, and Yao Mu. RoboTwin 2.0: A Scalable Data Generator and Benchmark with Strong Domain Randomization for Robust Bimanual Robotic Manipulation. In *Proceedings of the 43rd International Conference on Machine Learning*, 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.

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.

Erik Helmut, Niklas Funk, Tim Schneider, Cristiana de Farias, and Jan Peters. Tactile-Conditioned Diffusion Policy for Force-Aware Robotic Manipulation. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

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

Carolina Higuera, Akash Sharma, Taosha Fan, Chaithanya Krishna Bodduluri, Byron Boots, Michael Kaess, Mike Lambeta, Tingfan Wu, Zixi Liu, Francois Robert Hogan, and Mustafa Mukadam. Tactile Beyond Pixels: Multisensory Touch Representations for Robot Manipulation. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 105–123. 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:2507.09160, 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.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Luyao Zhang, Mingrui Yu, Zelin Gao, Nan Xue, Boyu Zhou, Xing Zhu, Mingyu Ding, Yujun Shen, and Yinghao Xu. Causal World Modeling for Robot Control. In *Proceedings of Robotics: Science and Systems*, 2026a.

Xukun Li, Yu Sun, Lei Zhang, Bosheng Huang, Yibo Peng, Yuan Meng, Haojun Jiang, Shaoxuan Xie, Guocai Yao, Alois Knoll, Zhenshan Bing, Xinlong Wang, and Zhenguo Sun. DECO: Decoupled Multimodal Diffusion Transformer for Bimanual Dexterous Manipulation with a Plugin Tactile Adapter. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026b.

Toru Lin, Yu Zhang, Qiyang Li, Haozhi Qi, Brent Yi, Sergey Levine, and Jitendra Malik. Learning Visuotactile Skills With Two Multifingered Hands. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5637–5643, 2025.

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.

Siyuan Ma, Boshi Zhang, Yutian Zhang, Qinglian Wu, Jiaqi Zhai, Dong Wei, and Qiaojun Yu. DECOWAM: decoupled whole-body world-action model for legged mobile manipulation. *arXiv preprint arXiv:2608.20114*, 2026.

Motubrain Team, Chendong Xiang, Fan Bao, Haitian Liu, Hengkai Tan, Hongzhe Bi, James Li, Jiabao Liu, Jingrui Pang, Kiro Jing, Louis Liu, Mengchen Cai, Rongxu Cui, Ruowen Zhao, Runqing Wang, Shuhe Huang, Yao Feng, Yinze Rong, Zeyuan Wang, and Jun Zhu. Motubrain: An Advanced World Action Model for Robot Control. arXiv preprint arXiv:2604.27792, 2026.

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

Shuai Tian, Yupeng Zheng, Yuhang Zheng, Songen Gu, Yujie Zang, Yuxing Qin, Weize Li, Haoran Li, Wenchao Ding, and Dongbin Zhao. VT-WAM: Visual-Tactile World Action Model for Contact-Rich Manipulation. arXiv preprint arXiv:2607.02503, 2026.

Hanwen Wang, Weizhi Zhao, Xiangyu Wang, Siyuan Huang, He Lin, Boyuan Zheng, Rongtao Xu, Gang Wang, Yao Mu, He Wang, Lue Fan, Hongsheng Li, Zhaoxiang Zhang, and Tieniu Tan. DexJoCo: A Benchmark and Toolkit for Task-Oriented Dexterous Manipulation on MuJoCo. arXiv preprint arXiv:2605.16257, 2026a.

Xiang Wang, Zhifei Zhang, He Zhang, Zhe Lin, Yuqian Zhou, Qing Liu, Shiwei Zhang, Yijun Li, Shaoteng Liu, Haitian Zheng, Jason Kuen, Yuehuan Wang, Changxin Gao, and Nong. Sang. Hbridge:H-shape bridging of heterogeneous experts for unified multimodal understanding and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14767–14778, 2026b.

Yihao Wang, Pengxiang Ding, Lingxiao Li, Can Cui, Zirui Ge, Xinyang Tong, Wenxuan Song, Han Zhao, Wei Zhao, Pengxu Hou, Siteng Huang, Yifan Tang, Wenhui Wang, Ru Zhang, Jianyi Liu, and Donglin Wang. VLA-Adapter: An effective paradigm for tiny-scale vision-language-action model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 18638–18646, 2026c.

Yuqi Wang, Xinghang Li, Wenxuan Wang, Junbo Zhang, Yingyan Li Li, Yuntao Chen, Xinlong Wang, and Zhaoxiang Zhang. Unified vision-language-action model. In *International Conference on Learning Representations*, 2026d.

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 (CVPR)*, pp. 26340–26353, 2024.

Angen Ye, Boyuan Wang Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Hao Li, Hengtao Li, Jie Li, Jindi Lv, Jingyu Liu, Min Cao, Peng Li, Qiuping Deng, Wenjun Mei, Xiaofeng Wang, Xinze Chen, Xinyu Zhou, Yang Wang, Yifan Chang, Yifan Li, Yukun Zhou, Yun Ye, Zhichao Liu, and Zhu Zheng. GigaWorld-Policy: an efficient action-centered world–action model. *arXiv preprint arXiv:2603.17240*, 2026.

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

Jialiang Zhao, Naveen Kuppuswamy, Siyuan Feng, Benjamin Burchfiel, and Edward Adelson. Poly-Touch: A robust multi-modal tactile sensor for contact-rich manipulation using tactile-diffusion policies. In *In 2025 IEEE International Conference on Robotics and Automation*, pp. 104–110, 2025.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Ya-Qin Zhang, Jiangmiao Pang, Jingjing Liu, Tai Wang, and Xianyuan Zhan. X-VLA: Soft-Prompted Transformer as Scalable Cross-Embodiment Vision-Language-Action Model. In *International Conference on Learning Representations*, 2026a.

Yuhang Zheng, Songen Gu, Yupeng Zheng, Weize Li, Yujie Zang, Shuai Tian, Xiang Li, Ce Hao, Chen Gao, Si Liu, Haoran Li, Yilun Chen, Shuicheng Yan, and Wenchao Ding. OmniVTA: Visuo-Tactile World Modeling for Contact-Rich Robotic Manipulation. arXiv preprint arXiv:2603.19201, 2026b.

Xinyue Zhu, Binghao Huang, and Yunzhu Li. Touch in the Wild: learning fine-grained manipulation with a portable visuo-tactile gripper. In *Advances in Neural Information Processing Systems*, volume 38, pp. 153783–153812, 2026.

A TACTILE DATA AND AUTOENCODER DETAILS

This appendix describes the tactile data sources and dataset splits, simulation-based tactile acquisition, canonical mapping and preprocessing, and tactile autoencoder implementation. It also reports the evaluation protocol and additional reconstruction results.

A.1 DATA SOURCES AND DATASET SPLITS

We train the Unified Tactile Autoencoder on five data subsets from RoboTwin, DexJoCo, and ManiFeel. RoboTwin contributes 27,500 trajectories from the Clean and Random configurations. DexJoCo contains 1,080 replayed trajectories and 900 additional trajectories generated through simulation-based trajectory augmentation. ManiFeel provides 200 trajectories with three-dimensional tactile observations across four insertion tasks.

Each subset is split by trajectory into approximately 80% training and 20% held-out evaluation data. All windows from the same trajectory remain in the same split, and the training subsets are combined to train a shared autoencoder.

A.2 SIMULATION-BASED TACTILE ACQUISITION

We record tactile signals during trajectory replay or execution in simulation. RoboTwin uses tactile grids on the four finger surfaces of its two grippers, whereas DexJoCo represents the palm and finger-joint surfaces with planar grids and the fingertips with curved grids. Each surface is discretized into an $H \times W = 10 \times 14$ grid with one normal and two tangential force components. A spatial mask $\mathbf{S}^{(r)}$ identifies valid nodes, and the corresponding sensing layouts are shown in Figure A1.

At each physics substep, RoboTwin provides contact impulses from the SAPIEN solver. For DexJoCo, MuJoCo contact forces are transformed to the world frame and multiplied by the physics timestep to obtain impulses. In both platforms, the impulse direction is adjusted to represent the forcing acting on the sensor. Contact positions and impulses are then expressed in the local frame of the corresponding sensing surface and projected onto its normal and two tangential directions. For curved fingertips, the projection basis is evaluated at the contact position.

We retain contacts with valid surface orientation and spatial location. RoboTwin filters contacts according to the surface normal, whereas DexJoCo additionally checks the designated palmar or fingertip regions and their geometric tolerance. Retained impulses are distributed to neighboring grid nodes using bilinear weights, renormalized over valid nodes according to $\mathbf{S}^{(r)}$. Contacts without valid neighboring nodes are discarded. For tactile frame k, the average force at node i on surface r is

$$\mathbf{f}_{k,i}^{(r)} = \frac{1}{\Delta t_k} \sum_{s \in \mathcal{I}_k} \sum_{c \in \mathcal{C}_s^{(r)}} w_{i,c,s}^{(r)} \mathbf{j}_{c,s}^{\mathrm{loc},r}, \tag{4}$$

where $\mathcal{I}_k$ denotes the physics substeps in the sampling interval, $\mathcal{C}_s^{(r)}$ contains the retained contacts, and $w_{i,c,s}^{(r)}$ is the normalized masked bilinear weight. The interval duration Δt_k includes substeps without contact. The output is a three-channel force field in newtons, without division by node area. The nominal sampling intervals are $60\,\mathrm{ms}$ for RoboTwin and $20\,\mathrm{ms}$ for DexJoCo. Spatial masks, actual interval durations, and empty observations are also recorded.



(a) RoboTwin (b) DexJoCo

Figure A1: **Tactile sensing layouts.** (a) Inner sensing surface of a RoboTwin gripper finger. (b) Palmar sensing surfaces of a DexJoCo hand. Colored areas indicate sensing coverage, and white dots denote grid nodes.

Table A1: **Fixed preprocessing parameters.** Channel tuples follow the order (n, u, v). Native units denote the original ManiFeel data scale; N denotes newtons.

Source	Input unit	Deadband thresholds $(\delta_{\text{lo}}, \delta_{\text{hi}})$	Asinh scales $(\kappa_n, \kappa_u, \kappa_v)$	Upper bounds (b_n, b_u, b_v)
ManiFeel	Native units	$(10^{-5}, 10^{-4})$	$(10^{-4}, 2{\times}10^{-5}, 10^{-5})$	$(0.003, 0.003, 0.0035)$
RoboTwin	N	$(10^{-4}, 10^{-3})$	$(0.01, 0.001, 0.001)$	$(50, 5, 6.1)$
DexJoCo	N	$(10^{-4}, 10^{-3})$	$(0.01, 0.001, 0.001)$	$(6.3, 3, 2)$

A.3 Canonical Mapping and Preprocessing

Canonical Mapping. Canonical regions are ordered by hand, with the left hand preceding the right. Following the DexJoCo hand asset, the base template for each hand contains ordered regions for the thumb, index, middle, and ring fingers, followed by a palm region. For five-finger embodiments such as Xynova Flex2, a corresponding little-finger region is inserted before the palm while preserving the same spatial mapping and masking scheme. Single-hand observations are assigned to the right-hand regions. The four RoboTwin gripper surfaces map to the left thumb, left index, right thumb, and right index, respectively, while the ManiFeel gripper surface maps to the right index. Each gripper surface is divided into four ordered 5×7 patches. DexJoCo surfaces map to their corresponding finger or palm regions; the fourth thumb region is unobserved. A canonical region is marked as observed if it contains at least one valid sensing node, regardless of whether contact is present. This preserves the distinction between unavailable measurements and observed regions with no contact.

Force Preprocessing and Contact Labels. We order the force channels as the normal and two tangential components. ManiFeel retains its native TacFF scale with the normal component replaced by its absolute value, whereas RoboTwin and DexJoCo use the forces obtained from simulation. Let $\bar{\mathbf{f}}_i$ denote the force vector at node i before dead-zone processing, and let S_i indicate whether the node is valid. Using thresholds $\delta_{\text{lo}} < \delta_{\text{hi}}$, we define

$$\gamma_i = \text{clip}\left(\frac{\|\bar{\mathbf{f}}_i\|_2 - \delta_{\text{lo}}}{\delta_{\text{hi}} - \delta_{\text{lo}}}, 0, 1\right), \qquad g_i = S_i \gamma_i^2 (3 - 2\gamma_i). \tag{5}$$

The soft label g_i is used both for contact supervision and as a smooth gate that suppresses low-magnitude forces. For each channel $c \in \{n, u, v\}$, we apply channelwise asinh compression, normalization, and clipping:

$$F_{i,c} = S_i \, \text{clip}_{\mathcal{R}_c}\left[\frac{\text{asinh}\left(g_i \bar{f}_{i,c}/\kappa_c\right)}{\text{asinh}\left(b_c/\kappa_c\right)}\right], \tag{6}$$

where $\mathcal{R}_n = [0, 1]$ and $\mathcal{R}_u = \mathcal{R}_v = [-1, 1]$. Contact nodes are identified by $g_i \geq 0.5$. The preprocessing parameters remain fixed during training and evaluation; values are listed in Table A1.

A.4 Autoencoder Implementation and Training

The Unified Tactile Autoencoder shares its surface encoder, region pooling modules, Transformer layers, and decoder across all tactile sources. It processes each frame independently, maps valid surface measurements to canonical region tokens, and reconstructs the corresponding force fields. Unobserved regions are excluded by the region mask and produce zero outputs. The architecture configuration is summarized in Table A2.

Each training sample contains a fixed-length sequence of consecutive frames from one trajectory. We sample ManiFeel, RoboTwin Clean, RoboTwin Random, DexJoCo Agent, and DexJoCo Replay with proportions of 15%, 10%, 30%, 20%, and 25%, respectively. Windows are further stratified by contact state, including no contact, contact onset or release, dynamic contact, and stable contact. During training, each sensing surface is multiplied by a gain sampled uniformly from $[0.95, 1.05]$, shared across the frames and channels of that surface, and then clipped to the valid input range. Reconstruction targets and contact labels remain unchanged.

We train the autoencoder with force reconstruction, contact supervision, temporal change matching, and latent regularization:

$$\mathcal{L}_{\text{AE}} = \mathcal{L}_{\text{force}} + \mathcal{L}_{\text{contact}} + \mathcal{L}_{\text{change}} + 0.5\,\mathcal{L}_{\text{latent}}. \tag{7}$$

Table A2: **Unified tactile autoencoder architecture.**

Component	Configuration
Surface CNN	Channels: $3 \to 64 \to 128$; input resolution: 10×14
Surface attention	Two window-attention blocks with four heads; window size 5×7 and shift $(2, 3)$ in the second block
Region pooling	Shared learned query with projection $128 \to 48$
Region attention	Two masked Transformer blocks with four heads; hidden dimension 48 and FFN dimension 192
Frame representation	34 canonical region tokens, each with dimension 48
Decoder	Four-head cross-attention, followed by two residual MLP blocks $(48 \to 192 \to 48)$ and convolutional refinement $(48 \to 64 \to 48)$

Table A3: **Tactile autoencoder reconstruction results.** Results on the combined held-out set. Contact F1 and Transition F1 are computed from predicted contact probabilities and their temporal changes, respectively. Force→Contact F1 derives contact states from reconstructed force fields. MAEs are reported in source-specific normalized units.

		Uniformly sampled windows				High-change windows		
Source	Windows	Contact F1 ↑	Transition F1 ↑	Force MAE ↓	Force→Contact F1 ↑	Contact F1 ↑	Transition F1 ↑	ΔForce MAE ↓
ManiFeel	80	99.69	54.22	0.0548	99.67	99.77	66.96	0.0690
RoboTwin Clean	1,030	96.38	88.41	0.0185	92.15	96.77	89.95	0.0185
RoboTwin Random	10,132	96.73	89.59	0.0174	92.81	96.57	89.50	0.0185
DexJoCo Agent	360	95.39	71.24	0.0189	88.27	91.55	74.44	0.0294
DexJoCo Replay	438	84.71	61.99	0.0372	75.13	83.56	63.01	0.0565

The four terms preserve force magnitude, contact support, temporal contact dynamics, and consistent representations for observed non-contact regions. We optimize the model with AdamW and mixed precision. After pretraining, the autoencoder is frozen and provides current tactile conditions, future tactile targets, and the decoder for the World Action Tactile Model.

A.5 Evaluation Protocol and Additional Results

We evaluate the frozen autoencoder using its EMA weights on held-out trajectories from all tactile data sources. For each trajectory, we sample uniformly distributed windows and high-change windows, each containing a fixed-length sequence of consecutive frames. High-change windows are selected according to the largest frame-to-frame change in normalized three-dimensional force within the window. No input augmentation is used during evaluation.

Invalid sensing positions are excluded from all metrics, while valid positions without contact are retained. Contact F1 compares binary ground-truth and predicted contact states using a threshold of 0.5. Transition F1 evaluates contact onset and release between adjacent frames at the same spatial locations. Contact force MAE measures the three-channel force reconstruction error at ground-truth contact positions, and transition delta MAE measures frame-to-frame force errors at contact onset and release. Force errors are computed in each source's normalized space, and metrics are averaged equally across the corresponding evaluation subsets.

Table A3 reports reconstruction results for each data subset. The contact branch achieves high Contact F1 on ManiFeel and RoboTwin, while the two DexJoCo subsets show different levels of difficulty. Transition F1 indicates that the representation captures contact onset and release, with lower scores on the more challenging subsets. Force-derived Contact F1 evaluates whether the reconstructed force fields preserve the contact distribution. All MAEs are computed in source-specific normalized units and should be compared within each data source.

Figure A2 provides qualitative examples of reconstructed force fields during contact onset, release, rapid force changes, and sustained contact.

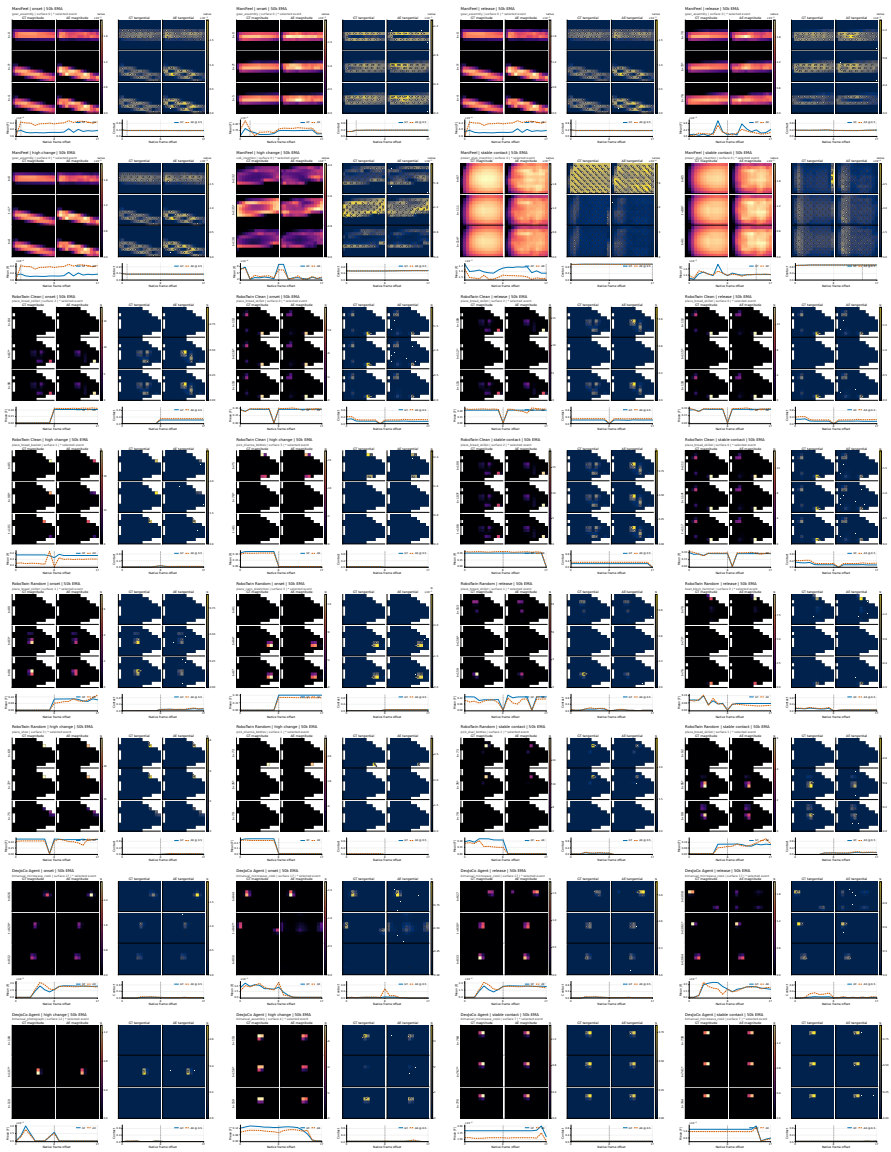

Figure A2: **Tactile reconstruction under different contact conditions.** Examples compare target and reconstructed force magnitudes and tangential force vectors during contact onset, release, rapid force changes, and sustained contact. Curves show mean force magnitude and contact fraction.

B Model and Experimental Details

This appendix provides implementation and experimental details omitted from the main text. We describe the model configuration, input organization, temporal alignment, training and inference procedures, and baseline-specific tactile conditioning.

B.1 Model Configuration

ME-Dex consists of three modality-specific Transformer experts for video, tactile, and action modeling. Each expert contains 30 Transformer blocks with its own input projection, hidden width, and output head. The Video, Tactile, and Action Experts use widths 3072, 1024, and 1024 on RoboTwin and DexJoCo, while the Tactile Expert uses width 512 on ManiFeel. Following H-Bridge, the blocks use an $8/14/8$ shallow–intermediate–deep allocation. The shallow and deep stages use modality-

specific self-attention. In the intermediate blocks $\mathcal{B}$, action and tactile features are projected from their native widths into a common 3072-dimensional attention space, before multi-modal joint self-attention and projection back to their modality-specific widths.

B.2 Input Organization and Temporal Alignment

Each sample combines current visual, tactile, and proprioceptive observations with noisy future video, tactile, and action targets. For RoboTwin and DexJoCo, a head or third-person view is placed above two wrist-camera views to form a composite image, which is resized and padded to 384×320 before visual encoding. Across all benchmarks, the model predicts 16 future action steps, 16 future tactile frames, and 8 future video frames. RoboTwin and DexJoCo sample action and tactile targets every three source frames and video targets every six frames. ManiFeel uses one wrist view and one tactile surface, with action and tactile targets sampled at every source timestep and video targets sampled every two timesteps.

B.3 Training and Inference Configuration

During training, each future modality is independently interpolated between its clean target and Gaussian noise using a uniformly sampled noise level. The three velocity-prediction losses use equal weights. The Video and Action Experts are initialized from pretrained Motus weights, the Tactile Expert is randomly initialized, and the Unified Tactile Autoencoder remains frozen. The mixed-training experiments use H-Bridge, whereas the Clean-only RoboTwin setting uses full multi-modal joint self-attention at all layers.

At inference, future video and tactile latents and the action chunk are initialized with Gaussian noise and jointly updated for ten flow-matching integration steps under fixed current observations. The resulting action chunk is denormalized and executed according to the protocol of each benchmark.

B.4 Baseline and Tactile-Conditioning Details

We follow official configurations whenever available. DP-T and DECO results are taken from their published benchmark releases, while $\pi_{0.5}$ and Fast-WAM are retrained with their official configurations. DECO.p applies DECO's released force–tactile adapter and checkpoint to our tactile-augmented DexJoCo data. All in-house runs use one random seed.

To study current tactile conditioning, we construct $\pi_{0.5}$+Tac, Fast-WAM+Tac, and ME-Dex (Tac Cond.) using the frozen Unified Tactile Autoencoder. For $\pi_{0.5}$+Tac, tactile tokens are concatenated with image and language tokens as prefix context for the Action Expert. For Fast-WAM+Tac and ME-Dex (Tac Cond.), they are concatenated with the T5 language-conditioning tokens. All other settings follow the corresponding baseline.

C Additional Experimental Results

C.1 RoboTwin Per-Task Results

To complement the aggregate results in the main text, we report success rates for all 50 RoboTwin tasks in Table C1. The results include ME-Dex, its tactile-conditioned and video-action variants, and the comparison policies under joint Clean–Random training. The final row reports the mean success rate across all tasks.

Under Random evaluation, ME-Dex improves over the video-action variant on 35 of 50 tasks, increasing the mean success rate from 86.90% to 91.92% (+5.02 percentage points). The Tac Cond. variant improves over the video-action variant on 26 tasks and raises the mean success rate by 2.64 percentage points. Compared with Tac Cond., the complete model improves on 33 tasks and adds another 2.38 percentage points.

We further report task-level results when training uses only Clean demonstrations. Table C2 compares ME-Dex with the methods listed on the RoboTwin leaderboard under Clean→Clean and Clean→Random evaluation.

Task	ME-Dex		ME-Dex (Tac Cond.)		ME-Dex (VA)		$\pi_{0.5}$		$\pi_{0.5}$ +TAC		Fast-WAM		Fast-WAM +TAC	
	C	R	C	R	C	R	C	R	C	R	C	R	C	R
Adjust Bottle	93	96	91	87	80	84	95	93	94	94	100	100	100	100
Beat Hammer	84	84	90	86	95	83	57	57	75	56	72	48	71	62
Rank Blocks (RGB)	92	95	92	87	91	82	59	75	47	67	97	94	100	98
Rank Blocks (Size)	74	82	65	75	66	74	28	43	25	30	87	84	89	83
Click Alarm	100	100	100	100	99	99	98	100	99	100	100	100	100	100
Click Bell	100	100	100	100	99	100	100	100	100	100	100	100	100	100
Dump Bin	98	98	97	96	96	88	74	87	77	87	95	97	96	92
Grab Roller	100	100	100	100	100	100	100	99	99	100	100	100	100	100
Handover Block	92	94	89	91	90	69	4	2	4	5	70	63	89	88
Handover Mic	100	98	100	95	98	95	58	60	66	61	61	88	65	89
Hanging Mug	45	53	37	45	23	32	7	7	5	9	31	50	39	43
Lift Pot	100	100	100	100	100	100	69	67	74	68	100	96	98	100
Move Can Pot	86	87	87	89	88	87	25	30	33	34	92	98	89	95
Move Pillbottle	97	100	96	100	96	97	21	33	21	31	99	100	100	100
Move Card	99	99	99	100	97	95	70	73	60	67	100	99	100	100
Move Stapler	83	79	87	80	83	71	9	15	14	16	71	64	62	71
Open Laptop	97	100	93	95	97	96	94	95	95	92	99	99	97	100
Open Microwave	90	78	81	76	20	27	65	70	79	74	89	40	93	37
Pick Diverse Bottles	87	89	82	89	89	85	21	42	22	46	93	92	88	90
Pick Dual Bottles	98	98	94	98	100	98	14	41	18	29	98	100	94	100
Place A2B Left	91	90	91	89	92	89	76	67	58	63	90	96	97	97
Place A2B Right	91	88	90	78	96	81	71	69	59	60	95	98	92	94
Place Bread Basket	91	87	92	93	90	92	39	54	35	58	91	93	87	90
Place Bread Skillet	89	92	90	90	85	90	26	43	27	35	89	91	93	89
Place Burger Fries	98	97	99	93	99	98	58	71	60	67	98	98	98	99
Place Can Basket	83	84	87	83	83	81	37	51	39	47	80	86	74	89
Place Cans	99	100	99	100	100	99	9	28	9	25	97	100	99	99
Place Container	99	100	100	97	99	99	91	93	82	94	97	99	100	96
Place Dual Shoes	94	94	86	88	79	79	31	51	38	52	92	96	91	91
Place Empty Cup	95	97	96	96	100	100	91	98	86	94	100	100	100	98
Place Fan	93	94	95	89	91	85	28	43	30	34	86	98	97	90
Place Mouse Pad	83	85	74	85	93	85	22	32	20	26	72	70	70	72
Place Object Basket	89	88	81	87	84	87	56	61	47	64	87	94	89	91
Place Object Scale	90	88	86	93	92	92	43	59	50	51	88	94	95	97
Place Object Stand	98	97	87	100	99	98	73	82	80	82	98	96	95	100
Place Phone Stand	88	86	86	78	80	70	36	51	14	27	83	77	85	84
Place Shoe	99	100	98	97	97	97	74	86	80	79	100	100	99	100
Press Stapler	100	99	98	98	97	100	97	96	96	97	95	96	98	97
Put Bottles	86	92	75	95	82	82	20	30	27	37	66	70	79	83
Put Object Cabinet	91	85	87	81	75	73	35	32	35	34	91	86	90	92
Rotate QR	90	81	84	76	85	85	31	34	49	30	80	79	82	83
Scan Object	76	81	68	79	81	76	13	21	10	14	83	80	79	71
Shake Bottle	100	99	99	97	100	97	97	100	98	97	100	100	100	99
Shake Bottle (Horiz.)	100	99	100	96	100	97	96	94	97	97	100	100	100	100
Stack Blocks (3)	98	95	90	86	95	87	20	40	14	27	99	97	97	96
Stack Blocks (2)	100	100	99	98	100	98	37	67	37	55	100	100	100	99
Stack Bowls (3)	82	89	80	88	81	85	60	70	56	60	85	88	83	90
Stack Bowls (2)	94	98	92	95	97	99	90	92	94	92	100	97	97	97
Stamp Seal	98	100	96	96	97	96	51	70	32	52	99	99	98	100
Turn Switch	78	81	66	67	81	86	62	63	64	67	66	71	77	80
Average	91.56	91.92	89.04	89.54	88.74	86.90	52.76	60.74	52.60	57.66	89.22	89.22	90.22	90.22

Table C1: **Per-task manipulation performance on RoboTwin under mixed training.** C and R denote Clean and Random evaluation, respectively. The models are trained jointly on Clean and Random trajectories. Task names are abbreviated for compactness.

ME-Dex achieves success rates of 88.7% and 76.2% under the two evaluation settings, respectively, corresponding to an average success rate of 82.5%. This exceeds the strongest listed baseline, OLA-Sem, by 11.2 percentage points in average success. ME-Dex also achieves the highest Clean→Clean success rate among the compared methods, while maintaining the highest Clean→Random success rate.

C.2 TACTILE AUTOENCODER PRETRAINING ABLATION

We compare two ME-Dex models that use tactile autoencoders pretrained on different data sources. One autoencoder is pretrained only on RoboTwin tactile data, whereas the other is pretrained jointly on the unified multi-source tactile data. Table C3 reports their closed-loop success rates under Random evaluation.

Using the unified multi-source tactile autoencoder improves Random success from 91.28% to 91.92%, a gain of 0.64 percentage points. This result shows that multi-source tactile pretraining does not degrade RoboTwin performance and provides a modest benefit for downstream manipulation.

Task	ME-Dex		Fast-WAM		$\pi_{0.5}$		4D-WAM		GigaBrain-0.7		OLA-Sem	
	C	R	C	R	C	R	C	R	C	R	C	R
Adjust Bottle	52	57	99	0	99	77	80	36	98	98	72	71
Beat Hammer	89	60	82	0	93	23	92	28	94	74	65	55
Rank Blocks (RGB)	95	71	81	0	72	44	88	10	68	76	64	66
Rank Blocks (Size)	70	69	52	0	45	21	54	8	42	28	36	37
Click Alarm	100	99	99	60	65	50	100	70	98	96	100	91
Click Bell	100	99	95	7	31	35	98	77	98	84	100	98
Dump Bin	97	82	94	0	92	84	86	57	90	66	96	94
Grab Roller	100	100	100	2	99	97	100	68	100	100	100	100
Handover Block	88	75	66	0	30	12	79	25	16	32	24	17
Handover Mic	96	94	98	0	98	6	85	13	100	100	89	80
Hanging Mug	37	24	14	0	17	14	70	10	26	36	16	15
Lift Pot	100	75	91	0	98	35	88	9	86	86	100	91
Move Can Pot	91	57	96	0	60	10	94	17	86	82	46	45
Move Pillbottle	96	83	92	0	46	32	89	22	32	58	78	71
Move Card	100	100	94	0	84	65	82	35	58	82	90	92
Move Stapler	83	52	33	0	20	18	43	12	26	40	58	53
Open Laptop	80	85	84	0	93	68	85	82	78	88	90	74
Open Microwave	89	25	17	0	86	59	68	12	82	50	76	43
Pick Diverse Bottles	83	63	72	0	66	42	77	44	38	50	76	77
Pick Dual Bottles	94	70	87	1	72	57	92	67	26	56	92	84
Place A2B Left	90	73	69	0	71	49	71	19	68	72	65	64
Place A2B Right	89	76	60	1	68	39	62	42	58	68	85	80
Place Bread Basket	93	92	87	0	63	60	87	77	60	68	84	80
Place Bread Skillet	91	82	85	0	76	47	82	69	80	50	81	67
Place Burger Fries	98	99	97	0	82	86	86	72	82	86	94	75
Place Can Basket	86	66	77	0	56	10	83	51	20	22	36	34
Place Cans	99	94	98	0	31	73	94	48	8	18	97	85
Place Container	99	86	100	0	92	72	93	82	94	94	97	92
Place Dual Shoes	85	30	67	0	70	45	66	19	68	72	87	76
Place Empty Cup	98	86	90	0	96	80	99	70	90	90	98	94
Place Fan	97	81	59	0	66	26	48	17	78	74	76	68
Place Mouse Pad	89	81	55	0	34	31	54	9	26	44	59	55
Place Object Basket	95	79	82	0	70	29	75	40	74	64	54	49
Place Object Scale	96	79	83	0	78	38	79	24	64	54	69	63
Place Object Stand	96	91	94	0	85	47	87	42	88	76	81	77
Place Phone Stand	79	71	85	0	65	37	84	21	36	52	75	63
Place Shoe	100	95	87	0	82	50	83	52	88	92	79	78
Press Stapler	99	97	90	3	79	63	76	61	90	74	98	95
Put Bottles	88	60	55	0	78	50	68	19	68	54	44	37
Put Object Cabinet	45	38	42	0	39	22	69	47	78	82	40	27
Rotate QR	86	68	81	0	90	21	79	67	54	58	46	40
Scan Object	84	69	68	0	49	40	73	53	32	46	42	32
Shake Bottle (Horiz.)	100	99	100	9	99	100	98	70	100	92	92	70
Shake Bottle	100	99	100	10	100	100	98	72	100	98	89	70
Stack Blocks (3)	90	70	89	0	77	25	95	82	60	78	74	52
Stack Blocks (2)	100	96	99	0	91	43	100	34	80	92	91	90
Stack Bowls (3)	82	77	80	0	81	52	82	27	70	64	88	81
Stack Bowls (2)	92	90	94	0	94	70	94	81	90	74	94	90
Stamp Seal	90	72	30	0	53	21	74	10	52	54	87	70
Turn Switch	59	74	39	3	52	25	86	11	42	52	86	70
Average	88.7	76.2	77.8	1.9	70.7	46.0	81.5	41.8	66.8	67.9	75.1	67.6

Table C2: **Per-task RoboTwin results under Clean-only training.** C and R denote Clean→Clean and Clean→Random evaluation, respectively. Baseline results are taken from the official RoboTwin leaderboard (Chen et al., 2026); task names are abbreviated for compactness.

Tactile AE pretraining data	Random success rate (%) ↑
RoboTwin only	91.28
Unified multi-source	**91.92**

Table C3: **Effect of tactile autoencoder pretraining data on RoboTwin.** Both models are evaluated under the same Random configuration.

C.3 Rollout Visualization with Tactile Data

We visualize ME-Dex rollouts together with camera observations and tactile signals in both simulated and physical environments. The examples show how the tactile observations change as the robot establishes, maintains, and releases contact during manipulation.

Figure C1 shows representative rollouts on DexJoCo and RoboTwin. The synchronized camera views and tactile maps capture contact changes during bimanual peg insertion and bread placement, covering both dexterous hands and grippers.

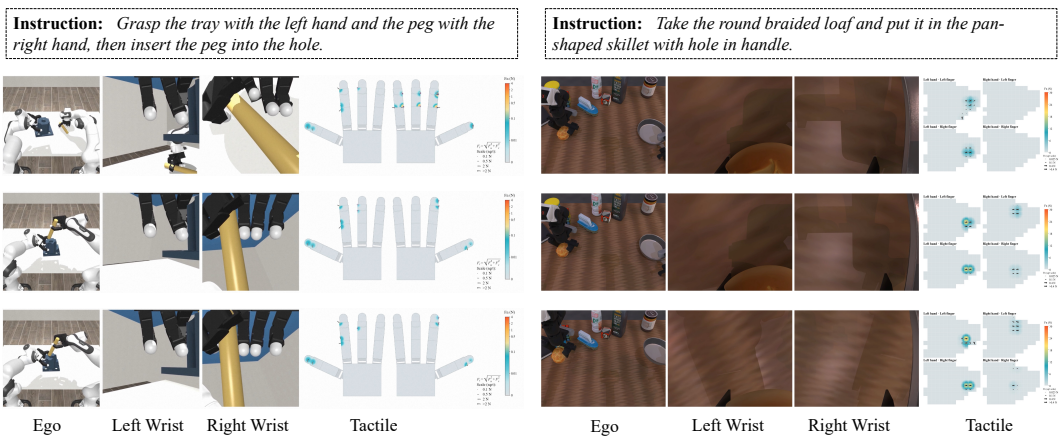

Figure C1: **Rollouts with tactile visualization in simulation.** Representative ME-Dex rollouts on DexJoCo (left) and RoboTwin (right). Each row shows synchronized camera views and tactile measurements at a different timestep. Time progresses from top to bottom.

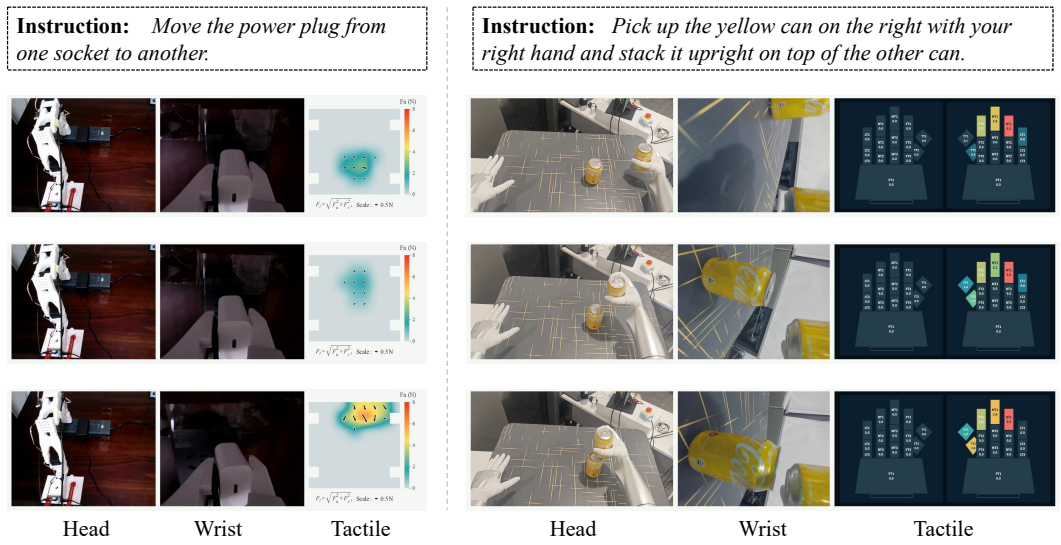

Figure C2: **Rollouts with tactile visualization on physical robots.** Representative ME-Dex rollouts on a LeRobot SO-101 with a Paxini PX6AX sensor (left) and Xynova Flex2 dexterous hands (right). Each row shows camera observations and tactile measurements at a different timestep.

Figure C2 shows tactile-aware rollouts on two physical platforms. The examples cover power-plug transfer with the LeRobot SO-101 and can stacking with the Xynova Flex2 hands. The tactile maps remain spatially interpretable across the different sensing layouts.

C.4 Cross-Modal Attention Visualization

Figure C3 visualizes attention from action tokens to video tokens for three RoboTwin tasks: Grab Roller, Open Microwave, and Place Bread Skillet. For ME-Dex, the attention changes across current and future video intervals as the action horizon advances. The maps tend to focus on the manipulated object and its contact region, such as the roller, the microwave handle, and the bread–skillet interaction. ME-Dex (VA) also accesses future video tokens, but its attention is generally more diffuse and less consistently localized around task-relevant regions. Fast-WAM can attend only to the current video frame $V(0)$ and therefore cannot attend to future video tokens. These visualizations suggest that ME-Dex learns both temporal correspondence and spatially focused visual representations for action generation.

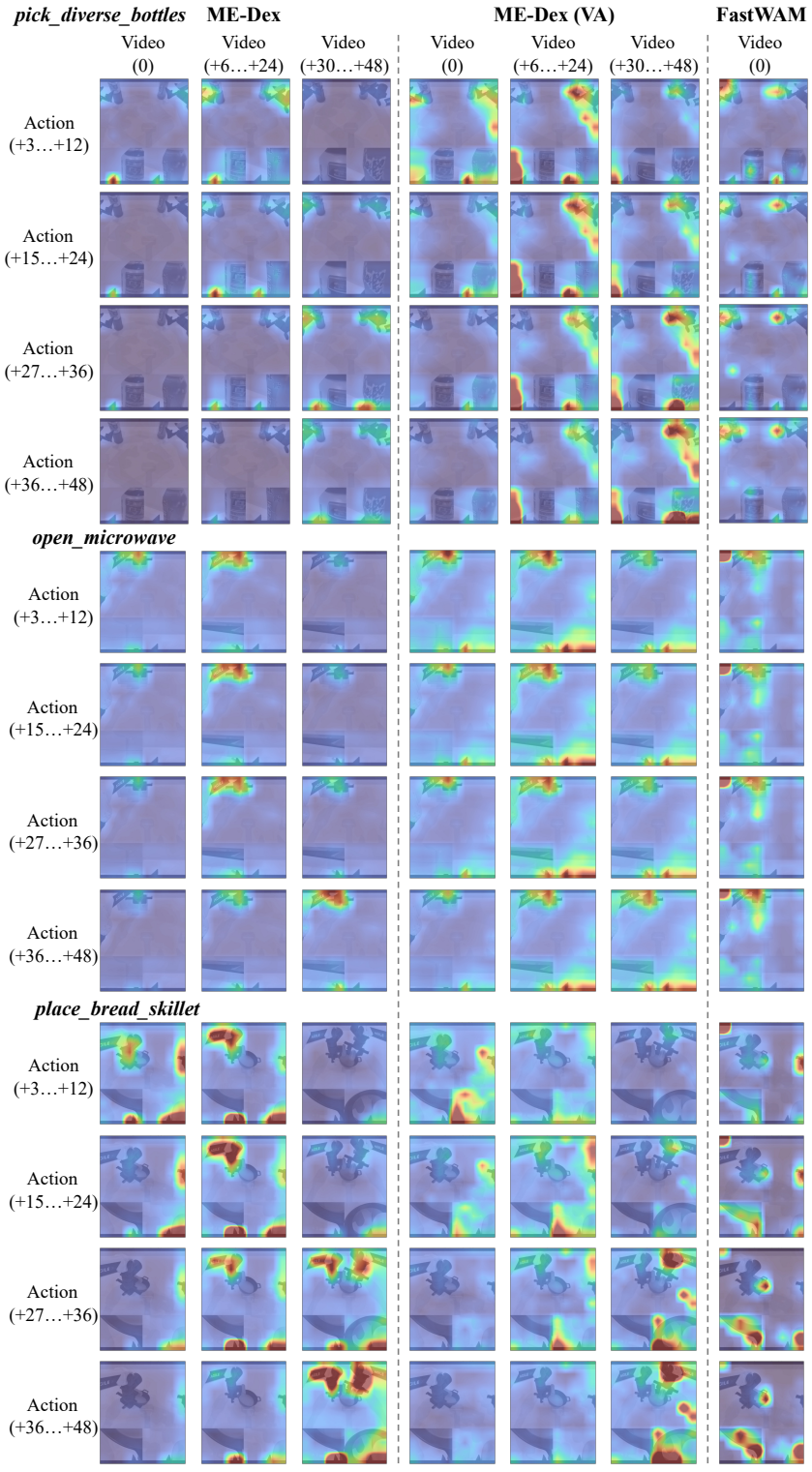

Figure C3: **Cross-modal attention maps on RoboTwin.** Attention from action tokens to video tokens at different temporal positions for three representative tasks. Rows correspond to action-token intervals, and columns correspond to current and future video intervals. The maps illustrate temporal correspondence and spatial concentration on task-relevant regions. Fast-WAM uses only the current video frame $V(0)$ in this comparison.

Table C4: Task descriptions and instructions for the LeRobot SO-101 PX.

Task	Instruction
`Upright_Medicine_Bottle`	Turn the plastic medicine bottle upright.
`Relocate_Power_Plug`	Move the power plug from one socket to another.
`Stack_and_Place_Box`	Stack the medicine bottles, then lift the stack and place it in the box.
`Object_Transfer_Box`	Move the tabletop objects into the box from left to right, one at a time.

Table C5: Task names and language instructions for the Xynova Flex2.

Task	Instruction
`Place_Can_Platform`	Pick up the yellow can with your right hand and place it upright on the white platform.
`Place_Tissues_Box`	Pick up the tissues on the table with your right hand and place them into the brown cardboard box on the right.
`Stack_Can`	Pick up the yellow can on the right with your right hand and stack it upright on top of the other can.

C.5 Real-Robot Evaluation

We evaluate ME-Dex on two physical platforms. The LeRobot SO-101 PX consists of a 5-DoF arm with one gripper actuation dimension. We design four manipulation tasks and collect 30 training demonstrations per task. The Xynova Flex2 platform comprises two 7-DoF arms and two dexterous hands with a combined 42 hand degrees of freedom, yielding 56 degrees of freedom in total. We design three manipulation tasks on this platform and collect 25 training demonstrations per task.

For each task, we first finetune the pretrained Unified Tactile Autoencoder on task-specific tactile data and then freeze it. ME-Dex is initialized from the RoboTwin-trained checkpoint and finetuned separately for each task using the adapted tactile representations. On the LeRobot SO-101 PX, we conduct 20 evaluation trials per task and compare against $\pi_{0.5}$ and Fast-WAM. On the Xynova Flex2, we report qualitative rollouts.

The four SO-101 PX tasks are `Upright_Medicine_Bottle`, `Relocate_Power_Plug`, `Stack_and_Place_Box` and `Object_Transfer_Box`. The three Xynova Flex2 tasks are `Place_Can_Platform` `Place_Tissues_Box`, and `Stack_Can`. Task names and language instructions are listed in Tables C4 and C5.

Quantitative Results. Table C6 reports success rates on the SO-101 PX. ME-Dex achieves an average success rate of 57.5% across the four tasks, exceeding $\pi_{0.5}$ by 17.5 percentage points and achieving a higher success rate on every task. Fast-WAM is evaluated on `Upright_Medicine_Bottle` and `Relocate_Power_Plug`, where ME-Dex improves the success rate by 30.0 and 35.0 percentage points, respectively.

Qualitative Results. Figure C4 compares representative rollouts on `Relocate_Power_Plug` (left) and `Upright_Medicine_Bottle` (right). These examples correspond to moving a power plug from one socket to another and turning a plastic medicine bottle upright. ME-Dex completes both tasks, whereas $\pi_{0.5}$ and Fast-WAM fail in the illustrated trials.

Table C6: **Quantitative comparison on the LeRobot SO-101 PX.** Success rates are reported over 20 evaluation trajectories per task. The Fast-WAM average is computed over the two tasks for which results are available.

Task	ME-Dex	$\pi_{0.5}$	Fast-WAM
`Upright_Medicine_Bottle`	**80.0**	65.0	50.0
`Relocate_Power_Plug`	**35.0**	10.0	0.0
`Stack_and_Place_Box`	**50.0**	30.0	–
`Object_Transfer_Box`	**65.0**	55.0	–
Average	**57.5**	40.0	25.0

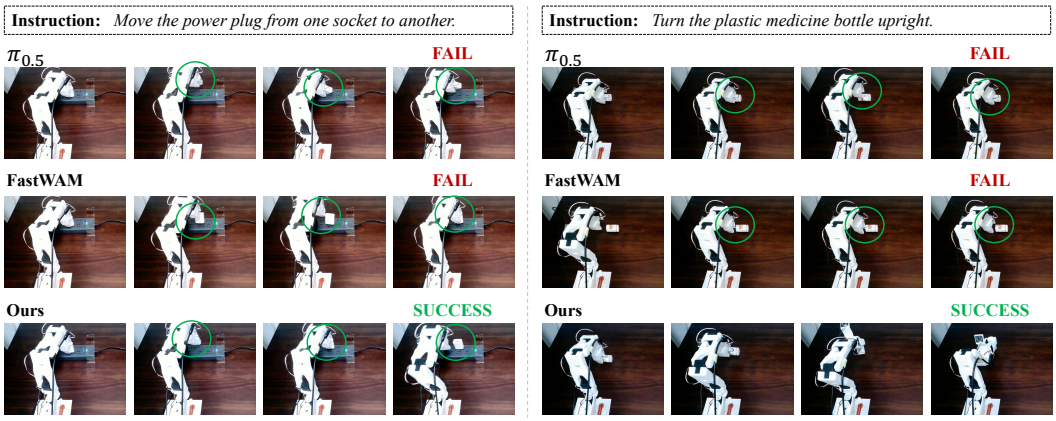

Figure C4: **Qualitative comparisons on the LeRobot SO-101 PX.** Representative rollouts of $\pi_{0.5}$, Fast-WAM, and ME-Dex on `Relocate_Power_Plug` (left) and `Upright_Medicine_Bottle` (right). ME-Dex completes both tasks, while the compared baselines fail in these examples. Time progresses from left to right within each row.

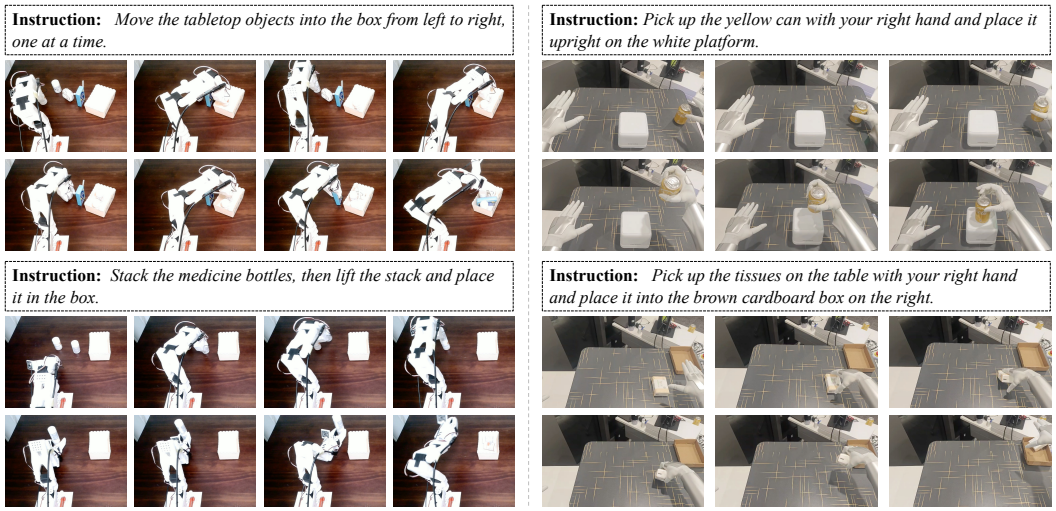

Figure C5: **Additional real-robot rollouts of ME-Dex.** Left: sequential object transfer (top) and bottle stacking and placement (bottom) on the LeRobot SO-101 PX. Right: `Place_Can_Platform` (top) and `Place_Tissues_Box` (bottom) on the Xynova Flex2. Within each task, frames progress from left to right and then from top to bottom.

Figure C5 shows additional ME-Dex rollouts. The left side presents the `Object_Transfer_Box` task (top) and the `Stack_and_Place_Box` task (bottom) on the LeRobot SO-101 PX. The right side presents two representative Xynova Flex2 tasks: `Place_Can_Platform` (top) and `Place_Tissues_Box` (bottom). These rollouts complement the quantitative evaluation with qualitative examples from both physical platforms. Additional rollout videos are provided in the supplementary material and at `https://anonymousmedex.github.io/`.

D The Usage of Large Language Models

We acknowledge the use of a large language model for limited assistance with manuscript writing and the implementation of the DexJoCo agent-based data generation pipeline. The model was used to proofread the manuscript, improve its linguistic fluency, and assist with code implementation. All AI-assisted code was reviewed and tested by the authors. The authors were solely responsible for the research ideas, methodology, experiments, data validation, analysis, and conclusions.