T-WLA: BRIDGING SLOW VISION AND FAST TOUCH

Anonymous authors
Paper under double-blind review

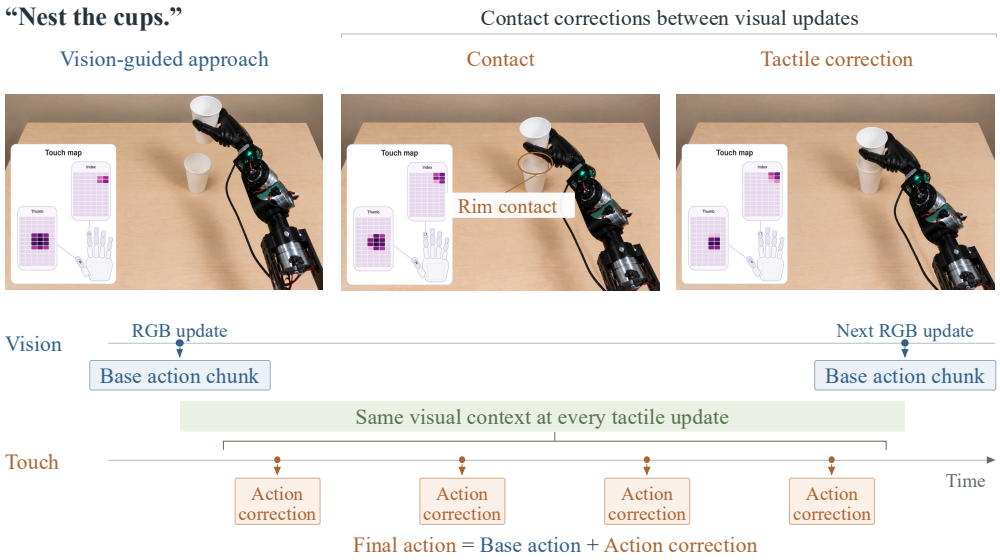


Figure 1: **Tactile World–Language–Action (T-WLA).** The visual branch uses current and past visual information to produce an action chunk and visual context. Between visual updates, the tactile editor uses this cached context and recent touch to correct the action chunk. During training, predicting future images and pressure helps the model learn features useful for control. Timing and event counts are illustrative. The visual and tactile examples are not synchronized.

ABSTRACT

Contact-rich manipulation requires tactile corrections guided by visual information and past observations. We introduce T-WLA, a world–language–action architecture that connects visual planning with tactile correction through cached visual context. At each visual update, the visual branch combines the current image with context from previous observations to generate an action chunk and update this context. Between visual updates, a lightweight tactile editor combines the cached context with recent tactile context to produce residual corrections without rerunning the visual branch. During training, future-image and future-pressure prediction help the model learn features useful for action generation and correction. We evaluate T-WLA on tactile-conditioned hand-motion prediction using EgoTouch and Tachin, and on four real-world dexterous manipulation tasks. T-WLA reduces finger-joint MPJPE by 30% relative to the strongest reported baseline and achieves an average success rate of approximately 50.0% across real-robot manipulation tasks. Videos and codes are available at the project website.

1 INTRODUCTION

Contact-rich manipulation refers to tasks whose success depends on precisely managing physical contact with objects or the surrounding environment (Hogan, 2022; Lee et al., 2019). Tactile feedback complements vision by revealing local contact conditions that are difficult to infer from images alone (Niu et al., 2026; Song et al., 2025; Xue et al., 2025; Zhou et al., 2026b). For example, when

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.

inserting a plug into a socket, vision guides alignment and approach, while touch provides feedback on contact and insertion resistance. Because these conditions can change between visual updates, tactile feedback supports rapid corrections to the ongoing motion.

Despite the importance of tactile feedback in contact-rich manipulation, many general-purpose robot policies rely primarily on vision. Vision–language–action (VLA) models adapt pretrained vision–language backbones to predict robot actions from visual observations and language instructions (Kim et al., 2024; Physical Intelligence et al., 2025). World–action models further integrate future-state prediction with action generation, linking control decisions to the anticipated evolution of the environment (Zhang et al., 2026b; Yuan et al., 2026). However, without tactile feedback, these models must infer local contact conditions indirectly from visual observations, which can limit their ability to respond to contact changes during precise manipulation.

Recent work incorporates tactile feedback into robot control through variable-rate expert architectures (Niu et al., 2026), residual adaptation (Wang et al., 2026), and inference-time policy steering (Wu et al., 2026b). We investigate a complementary question: *how can fast tactile feedback correct a slow visual policy between visual updates?* Our central idea is to generate asynchronous tactile corrections conditioned on persistent visual context. The visual branch processes RGB observations at 39.7 Hz, producing a base action chunk and a history-aware visual context at each update. The tactile branch processes touch at 119.1 Hz, combining recent tactile context with the cached visual context and the active action chunk to predict residual corrections to its remaining, unexecuted actions. This allows contact feedback to refine ongoing motion without waiting for the next visual update or rerunning the visual branch.

We instantiate this idea in T-WLA, a multi-rate architecture for asynchronous visual–tactile control (Fig. 1). The visual branch generates an action chunk and a history-aware visual representation, which is cached until the next visual update. Between these updates, a lightweight tactile editor combines the cached representation with recent tactile observations to produce high-frequency residual corrections to the ongoing action chunk. This separation allows the controller to respond to changing contact conditions without repeatedly running the visual branch. During training, future-image and future-pressure prediction regularize the control representations, encouraging them to capture information relevant to subsequent interactions. Our contributions are summarized as follows:

- **Asynchronous visuotactile control.** We propose T-WLA, a multi-rate architecture that caches history-aware visual representations to guide a lightweight tactile editor. The editor produces high-frequency residual corrections between visual updates without rerunning.
- **Predictive learning of control representations.** We regularize the visual representations through future-image prediction and the tactile features through both future-image and future-pressure prediction. These auxiliary objectives complement action supervision, encouraging the representations used for control to capture the evolving interaction.
- **Evaluation across human interaction and robot control.** We design an evaluation spanning tactile-conditioned hand-motion prediction and contact-rich real robot manipulation. Controlled comparisons and ablations assess whether retained visual history and predictive supervision improve tactile action correction.

2 Related Work

2.1 Memory and Action Chunk Execution

A single observation may not provide enough information to determine the next action. Memory-based policies address this limitation by retaining information from earlier observations; for example, MemoryVLA stores consolidated perceptual and cognitive entries that can be queried over long horizons (Shi et al., 2026). A complementary line of work predicts temporally extended action sequences. ACT introduced transformer-based action chunking to reduce the effective decision horizon (Zhao et al., 2023), while generative policies such as Diffusion Policy, RDT-1B, and π_0 generate multi-step actions with diffusion or flow-based objectives (Chi et al., 2025; Liu et al., 2025a; Black et al., 2024a). Such temporal coherence can reduce responsiveness when new observations arrive during execution (Liu et al., 2025b; Zeng et al., 2026). Bidirectional Decoding improves closed-loop adaptation by selecting among candidate chunks (Liu et al., 2025b), while Consistency Policy and

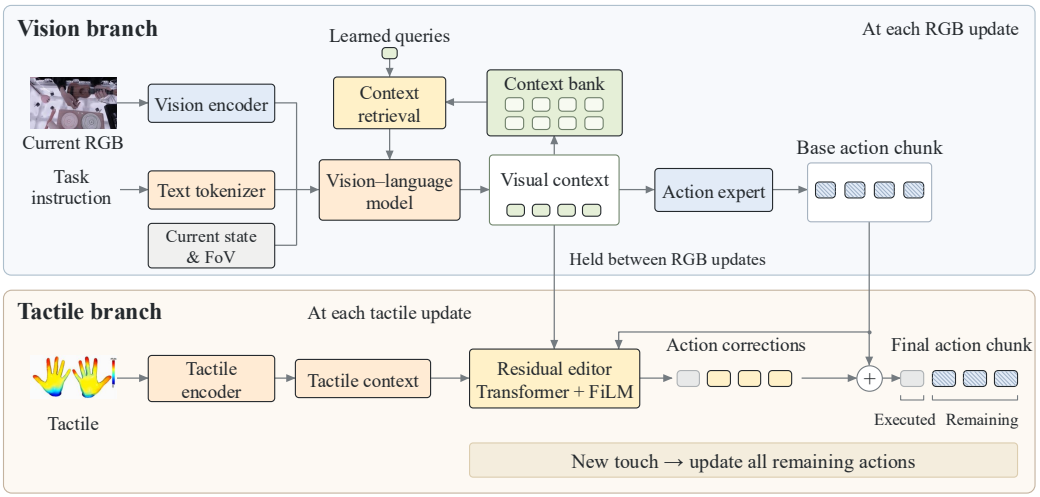


Figure 2: **Pipeline overview.** The visual branch processes RGB observations and past visual context to produce a base action chunk and updated visual context. Between visual updates, the tactile branch processes touch more frequently, using the cached visual context, tactile context, and base action chunk to predict residual corrections. These corrections update actions that have not yet been executed, allowing faster responses to contact without rerunning the visual branch.

Streaming Diffusion Policy accelerate generative action synthesis (Prasad et al., 2024; Høeg et al., 2025). Real-time chunking instead explicitly accounts for inference latency when transitioning between successive chunks (Black et al., 2025). T-WLA addresses a complementary setting: it retains visual history across policy updates and uses high-rate tactile feedback to revise the unexecuted portion of an active action chunk.

2.2 Tactile Foundation Models

Tactile representation learning aims to produce features that transfer across sensors and tasks. Heterogeneous Tactile Transformer learns a shared model across sensor types (Bi et al., 2026), while FG-CLTP aligns tactile features with visual and language representations (Ma et al., 2026). Other work explores transferring representations learned from human tactile recordings to dexterous robots (Zhang et al., 2026a) and scaling pretraining through tactile foundation models (Team et al., 2026; Zhou et al., 2026b). Our tactile encoder follows this direction through contrastive pretraining on synchronized tactile and hand-pose histories. Building on tactile representations, several systems study how to incorporate touch into manipulation policies. Multi-rate architectures allow tactile feedback to influence control between visual updates. T-Rex assigns vision and touch to experts operating at different rates (Niu et al., 2026), while Reactive Diffusion Policy combines a slow visual diffusion policy with a fast latent tactile controller (Xue et al., 2025). FA-RDP adapts this division to task demand (Zhuo et al., 2026), and DSWAM organizes world–action modeling into cooperating slow and fast systems (Zhu et al., 2026). Residual adaptation provides another mechanism for incorporating touch into existing policies. ViTaR attaches a lightweight visuotactile residual branch to a frozen backbone (Wang et al., 2026), while ResTacVLA applies residual adaptation in feature space (Zhang et al., 2026c). Tactile feedback can also guide action execution through inference-time adaptation. FBFM introduces asynchronous feedback during flow-matching execution without additional training (Li et al., 2026), while inference-time policy steering adjusts a fixed policy using vision and touch (Wu et al., 2026b). Complementary approaches use touch as training supervision without requiring tactile input at deployment (Gubernatorov et al., 2026; Cheng et al., 2026). These approaches enable tactile feedback to influence action generation and execution. In T-WLA, a cached, history-aware visual representation connects a slow visual policy to a fast tactile editor, which combines visual context with recent touch to refine the ongoing action chunk. Predictive supervision encourages these representations to capture evolving interaction dynamics, while auxiliary predictors are used only during training and add no inference cost.

3 METHODOLOGY

3.1 OVERVIEW

Contact-rich Language-conditioned manipulation is determined by low-frequency RGB observations I_t and higher-frequency tactile observations $T_{t,\tau}$, where t indexes visual updates and τ indexes tactile updates between consecutive visual observations. In this paper, we propose T-WLA which couples a visual policy with a lightweight tactile editor through a cached, history-aware visual representation. At each visual update, the visual branch integrates the current image, instruction, state, field-of-view and visual history to produce a base action chunk $A_t^{\text{base}} = \left(a_{t,0}^{\text{base}}, \ldots, a_{t,H-1}^{\text{base}}\right)$, and an updated representation M_{V}^t. Between visual updates, the tactile editor combines this cached representation with the base action chunk and recent tactile context to predict residual corrections $\Delta A_{t,\tau} = \left(\Delta a_{t,\tau,0}, \ldots, \Delta a_{t,\tau,H-1}\right)$ for unexecuted actions. During training, future-image and future-tactile prediction regularize the representations used for action generation and correction.

3.2 HISTORY-AWARE VISUAL REPRESENTATIONS

Multi-view and temporal observations provide complementary cues for understanding scene geometry and dynamics (Wang et al., 2025; Zhou et al., 2026c). Motivated by this insight, we augment the standard vision–language–action formulation (Kim et al., 2024; Brohan et al., 2023) with recurrent visual context. As illustrated in Fig. 2, the visual branch processes the current observation and language instruction together with a compact context of past observations. This context allows historical observations to inform current action prediction without repeatedly encoding the full observation history. At visual step t, the context bank $\mathcal{B}_{\text{V}}^{t-1}$ stores the last K sets of visual output context tokens. An attention-based context retrieval module $\mathcal{M}_{\text{V}}$ uses learnable query tokens Q_{V} to summarize this history into a fixed-size set of input context tokens:

$$M_{\text{V,in}}^t = \mathcal{M}_{\text{V}}\left(\mathcal{B}_{\text{V}}^{t-1}, Q_{\text{V}}\right). \tag{1}$$

Within $\mathcal{M}_{\text{V}}$, the queries attend to stored historical tokens together with learned reference tokens. The reference tokens are trainable parameters, whereas the historical tokens are saved outputs from previous visual steps. The retrieved context $M_{\text{V,in}}^t$ is then passed to the visual backbone alongside the current image features and language instruction. Neither the current image nor the current instruction enters this retrieval module as in Fig. 3 (a). The retrieved tokens are then processed jointly with current image tokens E_t, language tokens L_ℓ, and available state and field-of-view tokens c_t:

$$M_{\text{V}}^t = f_{\text{V}}\left(E_t, L_\ell, c_t, M_{\text{V,in}}^t\right), \tag{2}$$

$$A_t^{\text{base}} = f_{\text{act}}(M_{\text{V}}^t), \qquad \mathcal{B}_{\text{V}}^t = \text{Append}_K(\mathcal{B}_{\text{V}}^{t-1}, M_{\text{V}}^t). \tag{3}$$

The distinction between $M_{\text{V,in}}^t$ and M_{V}^t is essential: retrieval summarizes the past, while the final output incorporates the current observation and instruction. A_t^{base} denotes the base action chunk generated by the visual branch without tactile input. $\mathcal{B}_{\text{V}}^t$ denotes the updated history bank after storing the current visual representation M_{V}^t. Let $A_t^\star$ be the demonstration chunk aligned with A_t^{base}, and let $\bar{A}_t^\star = \text{Norm}(A_t^\star)$ be its standardized version. The action expert is trained by conditional flow matching (Lipman et al., 2022). For Gaussian noise ε and a sampled noise level $s \in [0, 1]$, define $X_s = (1 - s)\bar{A}_t^\star + s\varepsilon$. The action loss could be expressed as follow

$$\mathcal{L}_{\text{act}}^{\text{V}} = \mathbb{E}\left[\text{mse}_{\text{grp}}\left(v_\theta(X_s, s; M_{\text{V}}^t),\, \varepsilon - \bar{A}_t^\star\right)\right], \tag{4}$$

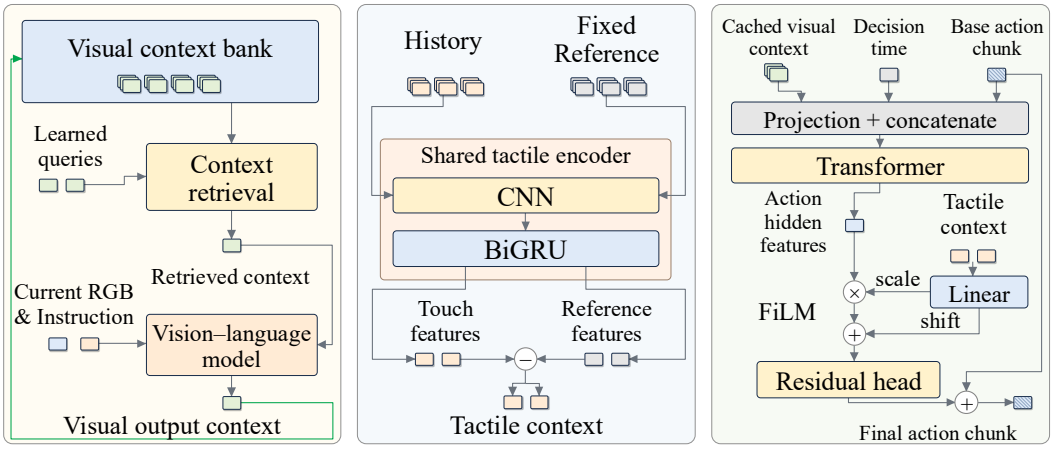
where v_θ is the action expert's velocity predictor. mse_{grp} averages squared errors over valid entries within each action-component group, then across the available groups.

3.3 ASYNCHRONOUS TACTILE RESIDUAL EDITING

The tactile branch captures recent contact evolution through a short observation window. At each tactile update, let $\mathcal{H}_{t,\tau}$ contain the eight most recent causally available tactile frames. A shared encoder processes the history of each hand to produce tactile features:

$$Z_{t,\tau} = \text{Enc}_{\text{T}}(\mathcal{H}_{t,\tau}), \tag{5}$$

where Enc_{T} denotes the complete tactile encoding pipeline, including sensor-validity masking and subtraction of the encoded fixed reference window as in Fig. 3 (b). There, at each tactile update,



(a) Visual Context **(b) Tactile Encoder** **(c) Residual Editor**

Figure 3: **Visual context, tactile encoding, and action correction.** (a) Learned queries retrieve past information from the visual context bank. The vision–language model combines this information with the current RGB image and instruction to produce updated visual context, which is stored for later use. (b) A shared tactile encoder processes recent tactile context and a fixed reference. Subtracting the reference features from the history features produces tactile context. (c) A Transformer combines the cached visual context, base action chunk, and decision time. The tactile context scales and shifts the resulting action features through FiLM. A residual head then predicts corrections that are added to the base action chunk.

tactile editor receives the cached visual context M_{V}^{t}, the base action chunk A_{t}^{base}, and the tactile-history features $Z_{t,\tau}$. It also receives $\Delta t_{t,\tau}$, the elapsed time since the active action chunk started, to identify the update's temporal position within the plan. The editor predicts a residual action chunk:

$$\Delta A_{t,\tau} = \mathcal{E}\big(M_{\mathrm{V}}^{t},\, A_{t}^{\mathrm{base}},\, Z_{t,\tau},\, \Delta t_{t,\tau}\big)\,, \tag{6}$$

where $\mathcal{E}$ denotes the complete tactile editor and $\Delta A_{t,\tau}$ has the same shape and action coordinates as A_{t}^{base}. Within the editor, learned projections first map visual-context and base-action tokens into a common feature space. Action-position and elapsed-time embeddings indicate where each command lies in the active plan. A small Transformer then combines the context and action tokens to produce visually contextualized action features. Tactile information enters after this step: each hand's tactile feature generates scaling and shifting coefficients that modulate these action features. Learned output layers map the modulated features to action corrections as in Fig. 3 (c). Missing tactile input produces no correction for that hand. Each update refines the cached base plan rather than accumulating corrections from previous updates, allowing contact feedback to adjust the remaining actions without rerunning the visual VLA. The corrected chunk is $\widehat{A}_{t,\tau} = A_{t}^{\mathrm{base}} + \Delta A_{t,\tau}$. We supervise it against the aligned demonstration $A_{t}^{\star}$ and penalize the residual magnitude:

$$\mathcal{L}_{\mathrm{edit}} = \mathbb{E}\Big[\mathrm{SmoothL1}_{\mathrm{grp}}\Big(\widehat{A}_{t,\tau}, A_{t}^{\star}\Big)\Big]\,, \tag{7}$$

$$\mathcal{R}_{\Delta} = \mathbb{E}[\mathrm{MeanSq}_{\mathrm{valid}}\,(\Delta A_{t,\tau})]\,. \tag{8}$$

We supervise the corrected action chunk with a group-balanced Smooth L1 loss $\mathcal{L}_{\mathrm{edit}}$ and regularize the magnitude of residual corrections $\mathcal{R}_{\Delta}$. The residual penalty discourages unnecessary deviations from the base plan by averaging squared corrections over valid, editable action entries. Appendix B provides the detailed architectures, reference construction, and validity masks for the visual context module and tactile editor.

3.4 Predictive Regularization of Control Representations

We use future-image and future-pressure prediction as auxiliary regularizers of the representations used for control. These objectives complement action supervision by encouraging the representations to capture interaction dynamics. Both predictors are used only during training.

Future-image prediction. The visual predictor conditions on current image tokens E_t and the final visual representation M_{V}^{t} to predict future image latents provided by a frozen image encoder. The

Table 1: Tactile-conditioned future hand-motion prediction. EgoTouch is reported across five native environments and as a macro-average; Tachin is evaluated on its held-out test split. Metrics use the same prediction horizon and are averaged per sequence before aggregation. Lower prediction error is better. Gray highlights T-WLA. **Bold**/underline indicate the best/second-best distinct reported errors within each metric and column, including ties. Tactile Freq. denotes the effective tactile-correction rate under our multi-rate protocol. Frequency is reported in Hz and is not ranked.

	EgoTouch						Tachin	
Method	Home	Workb.	Office	Retail	Outdoor	Avg.	Overall	Tactile Freq.
Wrist translation error (cm, ↓)								
VITRA + tactile	4.54	3.71	2.84	3.98	5.34	4.08	1.48	58.2
SmolVLA + tactile	4.55	3.74	2.88	3.98	5.33	4.10	1.50	69.0
MemoryVLA + tactile	4.65	3.78	2.96	4.08	5.41	4.18	1.55	15.0
T-Rex	4.63	3.76	2.95	4.05	5.41	4.16	1.54	17.16
T-WLA	**4.36**	**3.63**	**2.73**	**3.82**	**5.24**	**3.96**	**1.39**	**119.1**
Wrist rotation geodesic error (degrees, ↓)								
VITRA + tactile	6.29	6.22	4.81	6.86	7.15	6.27	4.55	58.2
SmolVLA + tactile	6.29	6.22	4.87	6.85	7.16	6.28	4.20	69.0
MemoryVLA + tactile	6.34	6.39	4.88	6.89	7.20	6.34	4.39	15.0
T-Rex	6.29	6.29	4.84	6.87	7.17	6.29	4.25	17.16
T-WLA	**6.18**	**5.83**	**4.79**	**6.66**	**7.03**	**6.10**	**4.17**	**119.1**
Wrist-relative finger-joint MPJPE (cm, ↓)								
VITRA + tactile	0.51	0.51	0.45	0.51	0.53	0.50	0.63	58.2
SmolVLA + tactile	0.51	0.50	0.46	0.50	0.53	0.50	1.28	69.0
MemoryVLA + tactile	1.37	1.37	1.17	1.34	1.46	1.34	1.87	15.0
T-Rex	1.17	1.13	1.07	1.17	1.18	1.14	1.08	17.16
T-WLA	**0.36**	**0.33**	**0.31**	**0.36**	**0.40**	**0.35**	**0.46**	**119.1**

resulting loss, $\mathcal{L}^{\mathrm{V}}_{\mathrm{vis}}$, regularizes the same visual representation used for action generation. During editor training, the visual predictor is frozen. A learned adapter incorporates tactile features $Z_{t,\tau}$ into its conditioning, yielding an additional objective $\mathcal{L}^{\mathcal{E}}_{\mathrm{vis}}$ for the tactile encoder. This adaptation affects only the prediction path, leaving the cached visual representation used for control unchanged.

Future-pressure prediction. The tactile predictor conditions on the cached visual representation, tactile features, and an upcoming window of edited actions to predict future pressure. Its objective, $\mathcal{L}_{\mathrm{tac}}$, regularizes the tactile features shared with the editor. Pressure targets use fixed training-set normalization, and gradients are stopped through the action condition.

Predictive objective. All three losses $\mathcal{L}^{\mathrm{V}}_{\mathrm{vis}}, \mathcal{L}^{\mathcal{E}}_{\mathrm{vis}}, \mathcal{L}_{\mathrm{tac}}$ use conditional flow matching. Let y denote a future image latent or normalized pressure target, and c the corresponding conditioning inputs. For Gaussian noise ε and a sampled noise level $s \in [0, 1]$, we define $y_s = (1 - s)y + s\varepsilon$ and minimize

$$\mathcal{L}_{\mathrm{pred}} = \mathbb{E}[\mathrm{mse}_{\mathrm{valid}}(\mathcal{W}(y_s, s; c),\, \varepsilon - y)]\,. \tag{9}$$

Future targets and their noisy versions enter only the auxiliary prediction paths, never the controller.

3.5 Training Strategy

Branch pretraining. We train the visual branch with the action objective $\mathcal{L}^{\mathrm{V}}_{\mathrm{act}}$ as in Eqn. 4 and the predictive regularizer $\mathcal{L}^{\mathrm{V}}_{\mathrm{vis}}$ as in Eqn. 9. Separately, we pretrain the tactile encoder on EgoTouch through contrastive learning between synchronized tactile and hand-pose histories. Let u_i and p_i be the normalized tactile and pose embeddings of a paired history, before reference subtraction on the tactile side. With similarities $S_{ij} = \alpha u_i^\top p_j$ and a learned scale $\alpha > 0$, the loss is

$$\mathcal{L}_{\mathrm{ctr}} = \tfrac{1}{2}\left[\mathrm{CE}_{\mathrm{diag}}(S) + \mathrm{CE}_{\mathrm{diag}}(S^\top)\right], \tag{10}$$

where $\mathrm{CE}_{\mathrm{diag}}$ is row-wise cross-entropy with the paired diagonal entry as the positive. Non-paired entries from the same recording are excluded from the denominator.

Tactile editor training. We freeze the visual branch and train the tactile encoder and editor using its cached memories and base actions. Training uses the corrected-action loss $\mathcal{L}_{\mathrm{edit}}$ as in Eqn. 7, the residual penalty $\mathcal{R}_\Delta$ as in Eqn. 8, and the predictive regularizers $\mathcal{L}^{\mathcal{E}}_{\mathrm{vis}}$ as in Eqn. 9 and $\mathcal{L}_{\mathrm{tac}}$ as in Eqn. 9. The visual predictor remains frozen, while its tactile conditioning adapter and the tactile predictor are trained.

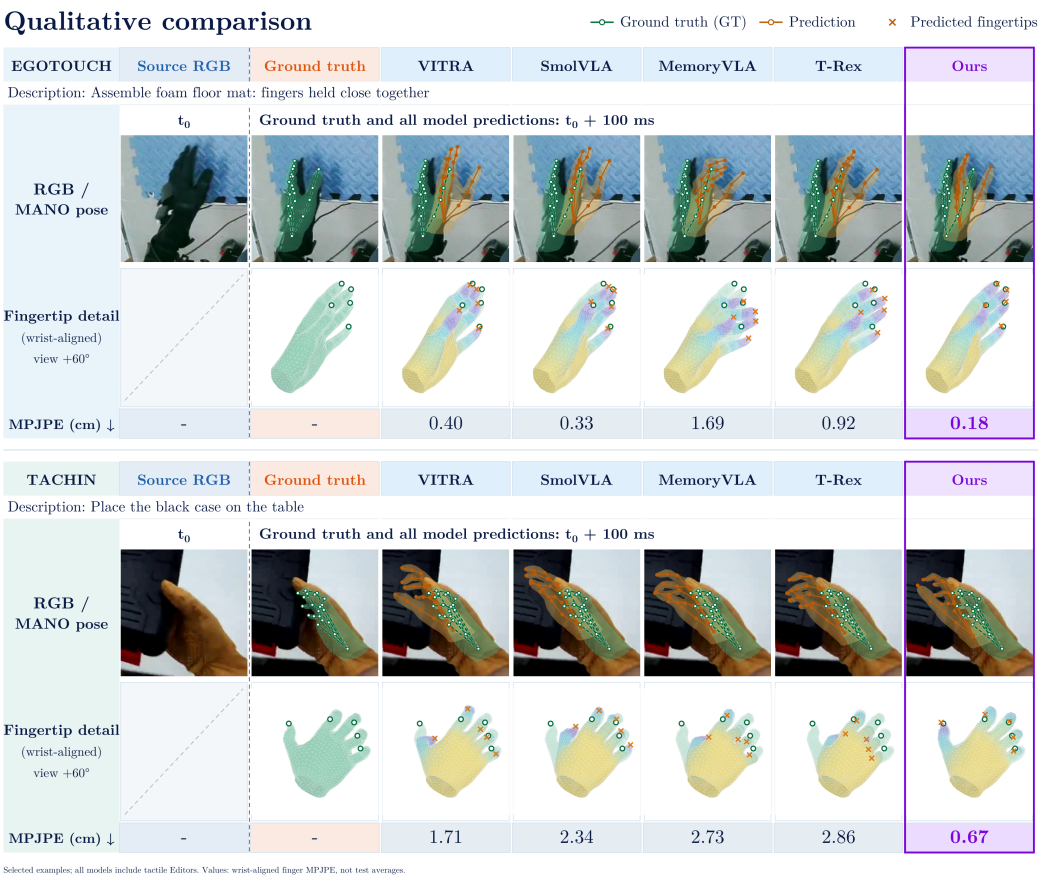


Figure 4: **Qualitative comparison on EgoTouch and Tachin.** VITRA, SmolVLA, MemoryVLA, and T-Rex denote the tactile-augmented variants used in Table 1. For each dataset, we show the source RGB frame at t_0, ground truth, and predictions from each of the models at $t_0 + 100\,\text{ms}$. RGB overlays depict ground-truth hand poses in green and predictions in orange. Fingertip-detail views show wrist-aligned hands rendered from a $+60°$ visualization view, with ground-truth hand mesh in green and predicted fingertips in orange crosses. MPJPE is reported in centimeters (↓: lower is better). T-WLA obtains the lowest MPJPE on both displayed examples.

Joint fine-tuning. Finally, we propose jointly fine-tuning both branches on EgoTouch using the above action and predictive objectives. Visual memories and base actions are recomputed online, and $\mathcal{L}_{\text{edit}}$ as in Eqn. 7 also updates the visual branch through the context interface. The visual predictor remains frozen. Both prediction heads are removed at inference. Appendix C.3 specifies loss reductions, normalization, gradient paths, and history construction during training.

4 Experiments

We organize our experiments around four questions. First, does combining visual context with high-rate tactile feedback improve future hand-motion prediction on human-interaction datasets (Section 4.2)? Second, does the tactile branch improve closed-loop success on real contact-rich robot tasks under a matched visual backbone and training budget (Section 4.3)? Third, which components—tactile input, recurrent memory, memory capacity, and predictive supervision—account for the improvement (Section 4.4)? Finally, does the visual branch remain a strong standalone predictor when tactile observations are unavailable (Appendix E)?

Table 2: Closed-loop success rate (%) on contact-rich robot tasks.

Controller	Egg	Sheet	Cup	Bag	Avg.
SmolVLA	0	28	0	39	16.75
SmolVLA + Tactile	0	30	1	37	17
T-WLA	18	72	**17**	82	47.25
T-WLA + Tactile	**21**	**78**	**17**	**84**	**50.0**

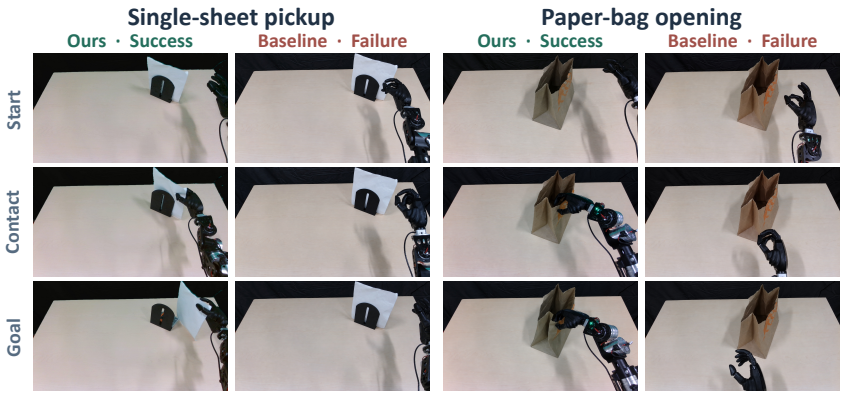


Figure 5: Representative real-robot trials for single-sheet pickup and paper-bag opening. Each column shows the start, contact, and goal frames of one episode; the left column of each task is a successful T-WLA trial and the right column is a failed baseline trial.

4.1 Experimental Setup

Datasets and tasks. For tactile-conditioned future hand-motion prediction, we evaluate on EgoTouch from TouchAnything (Zhou et al., 2026a) and the Tachin glove dataset (Tachintech, 2026). EgoTouch is evaluated in its five native environments—Home, Workbench, Office, Retail, and Outdoor—whereas Tachin is reported over its complete held-out test set. For visual-only prediction, we use GigaHands (Fu et al., 2025), OakInk2 (Zhan et al., 2024), and TACO (Liu et al., 2024). GigaHands is reported over its five official activity families, while OakInk2 and TACO are each evaluated over their complete held-out test sets. Our real-robot benchmark contains four contact-rich tasks: egg handling, single-sheet pickup, paper-cup nesting, and paper-bag opening.

Baselines and controlled comparison. For tactile-conditioned prediction, we compare with VITRA (Li et al., 2025), SmolVLA (Shukor et al., 2025), MemoryVLA (Shi et al., 2026), and T-Rex (Niu et al., 2026). When a baseline does not natively accept touch, its "+ tactile" variant receives the same synchronized tactile stream and uses the same task-specific tactile encoder, prediction head, and optimization budget as the other adapted baselines. T-Rex and T-WLA retain their native multi-rate interfaces.

4.2 Tactile- and Vision-Conditioned Action Prediction

Tactile-conditioned hand-motion prediction. Table 1 shows that T-WLA achieves the lowest reported errors across all three metrics in each of the five EgoTouch environments. Relative to the strongest reported baseline for each metric, it reduces macro-average wrist translation error, wrist rotation error, and finger-joint MPJPE by approximately 2.94%, 2.71%, and 30.00%, respectively. The largest improvement is in finger articulation: MPJPE decreases from 0.50 to 0.35 cm, while improvements in wrist translation and rotation are more modest. Compared with T-Rex, the corresponding reductions are 4.81%, 3.02%, and 69.30%. These improvements extend across all five environments, rather than being concentrated in a single category. On Tachin, T-WLA achieves the lowest reported error across all three metrics. Relative to the strongest baseline for each metric, wrist translation error decreases from 1.48 to 1.39 cm, wrist rotation error decreases from $4.20°$ to $4.17°$, and finger-joint MPJPE decreases from 0.63 to 0.46 cm, corresponding to reductions of approximately 6.08%, 0.71%, and 26.98%, respectively. As on EgoTouch, the largest improvement is in finger articulation, while the gains in global wrist translation and rotation are more modest. Across both datasets, the most consistent advantage of T-WLA is therefore improved prediction of fine-grained finger motion.

Table 3: Ablation of tactile input, memory, and predictive supervision. World-prediction ablations remove auxiliary objectives during editor training, not backbone pretraining. Lower is better for every metric. **Bold**/underline indicate the best/second-best distinct results in each column.

	EgoTouch			Tachin		
Variant	Trans. (cm)	Rot. (°)	Joint (cm)	Trans. (cm)	Rot. (°)	Joint (cm)
Without tactile input	4.010	6.200	0.430	1.450	4.400	0.500
Without recurrent memory	4.070	6.280	0.370	1.690	4.560	0.490
Shorter memory ($K_{\text{short}} < K$)	3.980	6.150	0.360	1.480	4.310	**0.460**
Without world prediction (T+V)	3.963	6.113	0.353	1.413	4.283	0.463
Without world prediction (T)	<u>3.962</u>	<u>6.112</u>	<u>0.352</u>	<u>1.412</u>	<u>4.263</u>	<u>0.462</u>
Without world prediction (V)	3.963	6.113	0.353	1.423	4.273	0.463
T-WLA	**3.960**	**6.100**	**0.350**	**1.390**	**4.170**	**0.460**

Appendix D.2 specifies the action representation, temporal alignment, metric definitions, and evaluation masks and aggregation.

4.3 Real-Robot Evaluation

Next, we test whether the prediction improvement translates into closed-loop manipulation. The four tasks are intentionally contact-sensitive: subtle slip can damage the egg, visual appearance alone does not reliably identify whether one or multiple sheets have been grasped, cup nesting requires regulating contact during insertion, and bag opening requires maintaining a deformable aperture. These properties make the benchmark a direct test of whether tactile feedback can correct an otherwise plausible visual action before failure becomes visually obvious. As shown in Table 2, T-WLA + tactile achieves a mean success rate of 50.0% across four real-robot tasks with only 30 teleoperated demonstrations per task. T-WLA, including its tactile editor, is trained only on human-hand data before robot adaptation. SmolVLA shares components with our visual branch and also achieves nonzero success with 30 demonstrations per task. VITRA and MemoryVLA achieve no successful trials under the same demonstration budget. Even without tactile input, T-WLA reaches 47.25% mean success, compared with 16.75% for SmolVLA. This gap is consistent with T-WLA's lower finger-joint prediction error (Table 1) and its use of visual history in base-action generation. Adding tactile input raises T-WLA's mean success from 47.25% to 50.0%, compared with an increase from 16.75% to 17.0% for SmolVLA. T-WLA gains 3 percentage points on egg handling, 6 on single-sheet pickup, and 2 on paper-bag opening, while paper-cup nesting remains at 17%. In observed SmolVLA failures, the hand often closes away from the target, leaving little contact feedback to guide a correction. This may explain why tactile input provides only a small gain when the base action fails to establish contact. Egg handling and paper-cup nesting remain the most challenging tasks, consistent with their more complex contact sequences.

4.4 Ablation Studies

Table 3 isolates the effects of tactile input, recurrent history, memory capacity, and predictive supervision under matched settings. Removing tactile input produces the largest increase in finger-joint error on both datasets: 22.9% on EgoTouch and 8.7% on Tachin. Translation and rotation errors also increase by 1.3% and 1.6% on EgoTouch, and by 4.3% and 5.5% on Tachin. Removing recurrent memory has a larger effect on wrist prediction, particularly on Tachin, where translation and rotation errors increase by 21.6% and 9.4%. Shortening memory from $K = 8$ to $K_{\text{short}} = 4$ increases error by 2.3% on average across the six metrics, while removing both world-prediction objectives increases it by 1.0% on the same basis. The full T-WLA achieves the lowest error in five of six metrics and ties for the lowest Tachin joint error.

5 Conclusion

We presented T-WLA, a multi-rate visuotactile architecture that caches history-aware visual context to guide high-frequency tactile corrections. A slow visual policy generates base action chunks, while a lightweight tactile editor reuses the cached context to refine ongoing actions at a higher frequency. Future-image and future-pressure prediction regularize the control representations during training, complementing action supervision with interaction dynamics. Extensive results show that joint visual-tactile modeling and predictive supervision improve contact-rich manipulation.

AI USE STATEMENT

Generative AI assisted with manuscript editing, mathematical exposition, literature checking, table formatting, and figure export. No experimental measurements were generated. The authors are responsible for verifying the method implementation, citations, and empirical claims.

REFERENCES

Jianxin Bi, Qiang Wang, Jayaram Reddy, Kelvin Lin, Soibkhon Khajikhanov, Ruihan Gao, and Harold Soh. Heterogeneous tactile transformer. *arXiv preprint arXiv:2606.29948*, 2026.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A Vision-Language-Action Flow Model for General Robot Control. *arXiv preprint arXiv:2410.24164*, 2024a.

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

Kevin Black, Manuel Galliker, and Sergey Levine. Real-Time Execution of Action Chunking Flow Policies. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-1122.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Ning Cheng, Jinan Xu, Wanlin Li, Yangzhi Chen, Jing Gao, Yiqun Wang, Kelan Peng, and Wenjuan Han. τ: Learning Touch-Augmented Vision-Language-Action Models from Future Visual Supervision. *arXiv preprint arXiv:2607.24485*, 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.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

Rao Fu, Dingxi Zhang, Alex Jiang, Wanjia Fu, Austin Funk, Daniel Ritchie, and Srinath Sridhar. Gigahands: A massive annotated dataset of bimanual hand activities. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17461–17474. IEEE, 2025.

Konstantin Gubernatorov, Mikhail Sannikov, Ilya Mikhalchuk, Egor Kuznetsov, Makar Artemov, Ogunwoye Faith Oluwatobi, Marcelino Fernando, Artem Asanov, Ziang Guo, and Dzmitry Tsetserukou. Hapticvla: Contact-rich manipulation via vision-language-action model without inference-time tactile sensing. *arXiv preprint arXiv:2603.15257*, 2026.

Sigmund Hennum Høeg, Yilun Du, and Olav Egeland. Streaming Diffusion Policy: Fast Policy Synthesis with Variable Noise Diffusion Models. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025.

Neville Hogan. Contact and physical interaction. *Annual Review of Control, Robotics, and Autonomous Systems*, 5(1):179–203, 2022.

Bohan Hou, Gen Li, Jindou Jia, Tuo An, Xinying Guo, Sicong Leng, Haoran Geng, Yanjie Ze, Tatsuya Harada, Philip Torr, et al. World model for robot learning: A comprehensive survey. *arXiv preprint arXiv:2605.00080*, 2026.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

Michelle A Lee, Yuke Zhu, Peter Zachares, Matthew Tan, Krishnan Srinivasan, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Jeannette Bohg. Making sense of vision and touch: Learning multimodal representations for contact-rich tasks. *arXiv preprint arXiv:1907.13098*, 2019.

Peize Li, Ruimeng Zhang, Ru Zhang, Cong Huang, Kai Chen, and Shanghang Zhang. Fbfm: A training-free asynchronous feedback mechanism for flow-matching in world-action models execution. *arXiv preprint arXiv:2607.29235*, 2026.

Qixiu Li, Yu Deng, Yaobo Liang, Lin Luo, Lei Zhou, Chengtang Yao, Lingqi Zeng, Zhiyuan Feng, Huizhi Liang, Sicheng Xu, et al. Scalable vision-language-action model pretraining for robotic manipulation with real-life human activity videos. *arXiv preprint arXiv:2510.21571*, 2025.

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

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. In *International Conference on Learning Representations*, volume 2025, pp. 29982–30009, 2025a.

Yuejiang Liu, Jubayer Hamid, Annie Xie, Yoonho Lee, Max Du, and Chelsea Finn. Bidirectional decoding: Improving action chunking via guided test-time sampling. In *International Conference on Learning Representations*, volume 2025, pp. 4594–4627, 2025b.

Yun Liu, Haolin Yang, Xu Si, Ling Liu, Zipeng Li, Yuxiang Zhang, Yebin Liu, and Li Yi. TACO: Benchmarking Generalizable Bimanual Tool-ACtion-Object Understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21740–21751, 2024.

Wenxuan Ma, Chaofan Zhang, Yinghao Cai, Guocai Yao, Shaowei Cui, and Shuo Wang. Fg-cltp: Fine-grained contrastive language tactile pretraining for robotic manipulation. *IEEE Robotics and Automation Letters*, 2026.

Dantong Niu, Zhuoyang Liu, Zekai Wang, Boning Shao, Zhao-Heng Yin, Anirudh Pai, Yuvan Sharma, Stefano Saravalle, Ruijie Zheng, Jing Wang, et al. T-rex: Tactile-reactive dexterous manipulation. *arXiv preprint arXiv:2606.17055*, 2026.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903. IEEE, 2024.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. pi0.5: A Vision-Language-Action Model with Open-World Generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Aaditya Prasad, Kevin Lin, Jimmy Wu, Linqi Zhou, and Jeannette Bohg. Consistency Policy: Accelerated Visuomotor Policies via Consistency Distillation. In *Proceedings of Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.071.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

Hao Shi, Bin Xie, Yingfei Liu, Lin Sun, Fengrong Liu, Tiancai Wang, Erjin Zhou, Haoqiang Fan, Xiangyu Zhang, and Gao Huang. Memoryvla: Perceptual-cognitive memory in vision-language-action models for robotic manipulation. In *International Conference on Learning Representations*, volume 2026, pp. 18567–18602, 2026.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, et al. Smolvla: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025.

Yuxin Ray Song, Jinzhou Li, Rao Fu, Devin Murphy, Kaichen Zhou, Rishi Shiv, Yaqi Li, Haoyu Xiong, Crystal Elaine Owens, Yilun Du, et al. Opentouch: Bringing full-hand touch to real-world interaction. *arXiv preprint arXiv:2512.16842*, 2025.

Tachintech. TachinTactileGlove-Demo01. Hugging Face dataset, 2026. Accessed September 23, 2026.

Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, et al. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*, 2025.

NeoteAI Team, Fudan TEAI Team, et al. N0-foundation: Towards the age of tactile intelligence, 2026.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5294–5306. IEEE, 2025.

Yi Wang, Renjun Wu, Jinyan Liu, and Xuesong Li. Vitar: Visuo-tactile residual adaptation for foundation vla manipulation. *arXiv preprint arXiv:2608.15816*, 2026.

Wei Wu, Fan Lu, Yunnan Wang, Shuai Yang, Shi Liu, Fangjing Wang, Qian Zhu, He Sun, Yong Wang, Shuailei Ma, et al. A pragmatic vla foundation model. *arXiv preprint arXiv:2601.18692*, 2026a.

Yilin Wu, Zilin Si, Zeynep Temel, Oliver Kroemer, and Andrea Bajcsy. Inference-time policy steering via vision and touch. *arXiv preprint arXiv:2606.14981*, 2026b.

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.

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

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

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-wam: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026.

Michael Zeng, Abhinav Agarwal, Ajay Bati, Brian Lee, Siddharth Ancha, and Russ Tedrake. Revisiting open-loop execution in robotics: Toward reactive, higher-performing policies. *arXiv preprint arXiv:2608.15938*, 2026.

Xinyu Zhan, Lixin Yang, Yifei Zhao, Kangrui Mao, Hanlin Xu, Zenan Lin, Kailin Li, and Cewu Lu. Oakink2: A dataset of bimanual hands-object manipulation in complex task completion. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 445–456. IEEE, 2024.

Chi Zhang, Penglin Cai, Ziheng Xi, Haoqi Yuan, Hao Luo, Wanpeng Zhang, Sipeng Zheng, Chaoyi Xu, and Zongqing Lu. Human-centric transferable tactile pre-training for dexterous robotic manipulation. *arXiv preprint arXiv:2607.01067*, 2026a.

Chushan Zhang, Jinguang Tong, Xuesong Li, Yikai Wang, and Hongdong Li. Keep the future, drop the rollout: Rift for world action models. *arXiv preprint arXiv:2608.11521*, 2026b.

Pengwei Zhang, Bin Xie, Xinpan Meng, Xinyu Guo, Ce Hao, Fang Deng, Long Cheng, and Tiancai Wang. Feeling the unexpected: Restacvla for contact-rich manipulation via residual tactile representation. *arXiv preprint arXiv:2607.03387*, 2026c.

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

Ruijie Zheng, Jing Wang, Scott Reed, Johan Björck, Yu Fang, Fengyuan Hu, Joel Jang, Kaushil Kundalia, Zongyu Lin, Loic Magne, Avnish Narayan, You Liang Tan, Guanzhi Wang, Qi Wang, Jiannan Xiang, Yinzhen Xu, Seonghyeon Ye, Jan Kautz, Furong Huang, Yuke Zhu, and Linxi Fan. FLARE: Robot Learning with Implicit World Modeling. In *Conference on Robot Learning (CoRL)*, 2025.

Jianyi Zhou, Ziteng Gao, Feiyang Hong, Zirui Liu, Guannan Zhang, Weisheng Dai, Ruichen Zhen, Chuqiao Lyu, Haotian Wu, Yinian Mao, et al. Touchanything: A dataset and framework for bimanual tactile estimation from egocentric video. *arXiv preprint arXiv:2605.13083*, 2026a.

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

Kaichen Zhou, Yuhan Wang, Grace Chen, Gaspard Beaudouin, Fangneng Zhan, Paul Liang, and Mengyu Wang. Page-4d: Disentangled pose and geometry estimation for vggt-4d perception. In *International Conference on Learning Representations*, volume 2026, pp. 36401–36414, 2026c.

Jian Zhu, Jianjun Zhang, Taiyi Su, Tianbin Liu, Zhangyuan Wang, Kai Xie, Zitai Huang, Chong Ma, Youzhang He, Tianjian Wang, et al. Dswam: A dual-system world action foundation model for fine-grained robot manipulation. *arXiv preprint arXiv:2607.04927*, 2026.

Lifeng Zhuo, Wendi Chen, Han Xue, Shirun Tang, Jun Lv, Cewu Lu, and Chuan Wen. Fa-rdp: A frequency-adaptive reactive diffusion policy for contact-rich manipulation. *arXiv preprint arXiv:2607.28596*, 2026.

APPENDIX

A Action Foundation Model **15**

B Module Implementation **15**

B.1 Visual Context Retrieval 15

B.2 Reference-Centered Tactile Encoding 16

B.3 Residual Editing and Contact Validity 16

C Learning and Execution Details **17**

C.1 Loss Reduction, Coordinates, and Flow Sampling 17

C.2 Predictive Conditions and Gradient Paths 18

C.3 History Construction and Cached Training 18

D Action and Evaluation Conventions **19**

D.1 Action Layout and Temporal Alignment 19

D.2 Evaluation Masks and Aggregation 19

E Visual-Only Action Prediction **20**

F Real-Robot Setup and Demonstration Collection **21**

G Additional Qualitative Results **21**

G.1 Additional EgoTouch Qualitative Examples 22

G.2 Additional Tachin Qualitative Examples 24

A Action Foundation Model

General-purpose manipulation policies increasingly adapt large pretrained backbones to action prediction (Wu et al., 2026a; Ye et al., 2026a; Yuan et al., 2026; Black et al., 2024b; Pertsch et al., 2025; Kim et al., 2025; Team et al., 2024; Liu et al., 2025a; Bjorck et al., 2025; Team et al., 2025; Shukor et al., 2025; O'Neill et al., 2024; Driess et al., 2023; Brohan et al., 2022; Reed et al., 2022). Vision–language–action (VLA) models follow this approach by predicting robot commands from visual observations and language instructions. RT-2 (Brohan et al., 2023) represents actions as tokens to jointly train on robot trajectories and web data, while OpenVLA (Kim et al., 2024) fine-tunes a pretrained vision–language backbone on a large collection of real robot demonstrations. More recent work expands the data and supervision available for policy learning. $\pi_{0.5}$ (Physical Intelligence et al., 2025) combines robot trajectories, web data, and subtask supervision to support generalization to unseen environments. SmolVLA (Shukor et al., 2025) emphasizes efficient training with community demonstrations, while VITRA (Li et al., 2025) learns a hand-motion prior from human activity videos. World–action models (WAMs) (Hou et al., 2026) further incorporate future-state prediction to learn representations of how the environment evolves. FLARE (Zheng et al., 2025) predict future observation embeddings rather than reconstructing future pixels. Dream-zero Ye et al. (2026b) jointly predict future video and robot actions. Fast-WAM (Yuan et al., 2026) use predictive learning during training while removing the predictor at deployment. These methods motivate learning control representations through future prediction. However, they run on single clock: visual observations, learned representations, and action predictions are updated together. In contact-rich manipulation, this coupling is restrictive because local contact conditions can change faster than the visual policy is updated. In these situations, a further challenge is to make these representations useful for tactile corrections between visual updates. T-WLA addresses this challenge by caching a history-aware visual representation that a faster tactile controller can reuse.

B Module Implementation

This appendix expands the implementation of the operators introduced in the main text. We retain M_{V}^{t} for the final visual context, $Z_{t,\tau}$ for the reference-centered tactile context, and $\Delta A_{t,\tau}$ for the effective action correction. Numerical specifications below refer to the reference implementation; the contact-filtered variant is identified explicitly where it changes an operation or its valid supervision set.

B.1 Visual Context Retrieval

Queries, reference slots, and historical tokens. The retrieval module uses $S = 8$ learned queries of width $d = 960$ and retains at most $K = 8$ sets of historical output context tokens. The query stream is initialized with Q_{V}. The reference slots in the key/value stream share this same parameter rather than introducing independent reference vectors. For slot j and a stored context of age k, the key/value inputs are

$$r_j = Q_{\mathrm{V},j} + e_j^{\mathrm{slot}}, \tag{11}$$

$$b_{k,j}^{t} = M_{\mathrm{V},j}^{t-k} + e_j^{\mathrm{slot}} + e_k^{\mathrm{age}}, \qquad k \in \{1, \ldots, K\}. \tag{12}$$

The queries attend to the concatenation of the reference slots and valid historical tokens. Slot and age embeddings are added at retrieval, not persistently written into the bank. Age counts visual updates rather than flow-integration steps or tactile events.

Masking and output extraction. Unused bank positions are zeroed and excluded by the attention mask; reference slots remain valid even when the bank is empty. Two retrieval blocks use pre-normalized cross-attention and SwiGLU feed-forward layers, with 15 attention heads and an intermediate width of 2560. The blocks use Q/K RMS normalization, a learned attention-logit scale, and zero-initialized $\tanh$ gates on their residual branches. These network gates are distinct from the pressure-based contact gate in Appendix B.3.

The retrieved input context is placed after the image, field-of-view, language, and state tokens in the visual backbone input. The final M_{V}^{t} is read directly from the last decoder layer at these eight context

positions, without an additional readout projection. The bank stores this observation-contextualized output, not the retrieved input context. Episode boundaries reset the bank; a visual event writes one new context set, independently of the number of action-flow integration steps.

B.2 Reference-Centered Tactile Encoding

Sensor processing and temporal encoding. Each hand is represented by its eight most recent causally available 21×21 pressure maps. Invalid temporal observations are zeroed, and the left-hand maps are flipped along the spatial horizontal axis to establish a common hand convention. Pressure and the spatial sensor-validity mask form two input channels. A shared CNN uses three 5×5 convolutional layers with 32 channels, padding 2, ReLU activations, and 2×2 max-pooling. The resulting 128-dimensional frame features enter a two-layer bidirectional GRU with hidden width 120 per direction. The final forward and initial backward states are concatenated and projected from 240 to 64 dimensions, followed by ℓ_2 normalization. Both GRU directions operate within the observed history window; neither accesses a future tactile frame.

Reference subtraction. Let F_{T} denote this shared, normalized encoder before reference subtraction. For hand $r \in \{\mathrm{L}, \mathrm{R}\}$, the complete encoding operator in the main text produces

$$Z^r_{t,\tau} = v^r_{t,\tau} \left[F_{\mathrm{T}}(\mathcal{H}^r_{t,\tau}) - F_{\mathrm{T}}(\mathcal{H}^r_{\mathrm{ref}}) \right]. \tag{13}$$

The validity indicator $v^r_{t,\tau}$ requires a valid encoded history and a valid most recent observation. In the contact-filtered variant, contact validity is also incorporated through the temporal mask. Each term in brackets is normalized before subtraction; their difference is not normalized again.

The fixed reference is a raw pressure window, not a cached embedding. Actual and reference windows are concatenated along the batch dimension, encoded together on every forward pass, and separated before subtraction. Consequently, both encoding paths receive gradients when the encoder is trained. The reference-statistics routine pools valid training pressure observations from history windows and the associated future-pressure targets, and repeats the per-taxel mean across the eight reference frames. These are fixed training-set statistics, not sample-specific future inputs to the controller. The exact statistics asset belongs to the training configuration.

B.3 Residual Editing and Contact Validity

Action features and tactile modulation. The editor maps the visual context and base-action tokens to width 256. Visual tokens use RMS normalization followed by a bias-free projection; base-action tokens are standardized for their input projection. Modality, action-position, and elapsed-time embeddings distinguish the token roles. The elapsed-time projection receives $10\Delta t_{t,\tau}$, with elapsed time expressed in seconds. A two-layer pre-normalized Transformer uses eight heads, a feed-forward width of 1024, GELU activations, and zero dropout. Only its action-position outputs, denoted H_{act}, enter the residual readout.

For each hand, a bias-free linear map from its 64-dimensional tactile context produces the FiLM scale and shift. With U^r denoting an internal, unmasked correction, the readout is

$$U^r = W^r_{\mathrm{out}}[\gamma^r(Z^r) \odot H_{\mathrm{act}} + \beta^r(Z^r)], \tag{14}$$

$$\Delta A_{t,\tau} = \sum_{r \in \{\mathrm{L},\mathrm{R}\}} m^r_{\mathrm{edit}} \odot U^r. \tag{15}$$

Here the affine modulation uses $\gamma \odot H_{\mathrm{act}} + \beta$, not $(1 + \gamma) \odot H_{\mathrm{act}} + \beta$. Each output projection is bias-free and zero-initialized. The mask m^r_{edit} intersects the editable action positions, that hand's component mask, and hand validity. Thus $\Delta A_{t,\tau}$ retains its main-text meaning as the effective correction added to the cached base chunk, rather than the raw network output. Every update is relative to the same base chunk, not to a previously corrected chunk.

Causal contact gate. The contact-filtered variant uses a hard gate computed from observed pressure history. Let $P_{j,r}$ be the pressure map of hand r at history index j, let $\mathcal{S}_r$ contain its valid sensor

sites, and let $\mathcal{J}_r$ contain its valid history indices. The gate is

$$c^r = \mathbf{1}\left[\max_{j\in\mathcal{J}_r} \frac{1}{5} \sum_{u\in \mathrm{Top5}(P_{j,r}|_{\mathcal{S}_r})} P_{j,r,u} \geq \eta\right], \qquad \eta = 0.25. \tag{16}$$

Here Top5 returns the indices of the five largest valid pressure values. An empty valid history gives $c^r = 0$. The reference threshold is expressed in the input pressure scale. The gate detects whether a valid history frame crosses the threshold; it does not require contact at the latest frame and is not learned from future contact labels.

The same c^r is intersected with the hand's temporal tactile-validity mask, action-supervision mask, and future-pressure-validity mask. A gated-out hand therefore receives zero correction and retains its base action, but its excluded action entries are not counted as zero-error observations in the loss. The contact wrapper does not directly change the RGB target validity mask. Sample-level contact filtering and its evaluation denominator are described in Appendix D.2.

C Learning and Execution Details

C.1 Loss Reduction, Coordinates, and Flow Sampling

Visual action loss. The group-balanced operators for the visual expert and the tactile editor have different reduction orders. Let b, j, and d index a minibatch sample, an action position, and an action component. For visual velocity error e_{bjd}, validity mask m^{V}_{bjd}, and component group $\mathcal{D}_g$, define

$$n_g = \sum_{b,j} \sum_{d\in\mathcal{D}_g} m^{\mathrm{V}}_{bjd}, \qquad \ell_g = \frac{\sum_{b,j} \sum_{d\in\mathcal{D}_g} m^{\mathrm{V}}_{bjd} e^2_{bjd}}{n_g}, \tag{17}$$

$$\widehat{\mathcal{L}}^{\mathrm{V}}_{\mathrm{act}} = \frac{\sum_{g:n_g>0} w_g \ell_g}{\sum_{g:n_g>0} w_g}. \tag{18}$$

Here ℓ_g is defined only for $n_g > 0$; a minibatch with no valid groups contributes zero loss. The within-group mean is formed over the entire minibatch, not separately for each sample. Empty groups are excluded. The reference configuration uses equal group weights and no additional horizon weighting. The six groups are translation, rotation, and articulation for each hand (Appendix D.1).

Edited-action loss and residual penalty. For corrected-action error $e_{bjd} = \widehat{A}_{bjd} - A^\star_{bjd}$, let $m^{\mathcal{E}}$ denote the effective valid, editable supervision mask and $n_{bg} = \sum_j \sum_{d\in\mathcal{D}_g} m^{\mathcal{E}}_{bjd}$. Write $\mathcal{G}_b = \{g : n_{bg} > 0\}$ and $\mathcal{B}_+ = \{b : |\mathcal{G}_b| > 0\}$. The minibatch estimator is

$$\widehat{\mathcal{L}}_{\mathrm{edit}} = \frac{1}{|\mathcal{B}_+|} \sum_{b\in\mathcal{B}_+} \frac{1}{|\mathcal{G}_b|} \sum_{g\in\mathcal{G}_b} \frac{\sum_j \sum_{d\in\mathcal{D}_g} m^{\mathcal{E}}_{bjd} \rho(e_{bjd})}{n_{bg}}, \tag{19}$$

where $\rho(e) = \frac{1}{2}e^2$ for $|e| < 1$ and $\rho(e) = |e| - \frac{1}{2}$ otherwise. Thus the editor averages nonempty groups within each sample before averaging supervised samples. Invalid targets are replaced safely before elementwise evaluation so that invalid entries do not introduce non-finite losses.

In contrast, the residual penalty is a single mean over all effective action entries:

$$\widehat{\mathcal{R}}_\Delta = \frac{\sum_{b,j,d} m^{\mathcal{E}}_{bjd} (\Delta A_{bjd})^2}{\sum_{b,j,d} m^{\mathcal{E}}_{bjd}}. \tag{20}$$

It does not rebalance groups or hands. Empty supervision contributes no action or residual loss. Contact filtering changes the valid sets in both expressions, rather than adding artificial zero-loss examples.

Coordinate normalization. The visual expert standardizes each action channel with fixed statistics, $(a_d - \mu_d)/\sigma_d$, and inverts this transform after sampling. The standard deviation is lower-bounded by 10^{-6}. For the editor, the corrected chunk, demonstration target, and correction share the action coordinates of the supplied cache. Any cache-level transformation precedes the operators above; the unified editor objective applies no additional standardization to its prediction, target, or residual. Normalizing the base-action input to the editor's feature projection is a separate operation and does not redefine the loss coordinates.

Noise levels and integration. The visual action-flow sampler draws $u \sim \mathrm{Beta}(1.5, 1)$ and sets $s = 0.001 + 0.999u$. During training, future-state flow objectives use uniform noise levels in $(0, 1)$ instead. Both retain the main-text interpolation from clean data at $s = 0$ to Gaussian noise at $s = 1$, with velocity target $\varepsilon - y$. Visual action inference uses ten Euler steps from $s = 1$ to $s = 0$ in the reference configuration. The final visual context is computed once and reused throughout this integration.

C.2 Predictive Conditions and Gradient Paths

Visual-branch prediction. The visual prediction condition combines current image tokens with the final output context, not the pre-backbone retrieval tokens. Future image latents are supplied by a frozen image encoder. Gradients from $\mathcal{L}^{\mathrm{V}}_{\mathrm{vis}}$ reach the current visual context; they do not backpropagate through detached history construction. Valid future horizons are reduced separately and then combined using their configured horizon weights.

Tactile-conditioned visual prediction. During cached editor training, a learned adapter maps $Z_{t,\tau}$ to an additive update of the eight context-derived rows of the visual predictor's condition. This update is confined to the prediction path and leaves M^t_{V} used for control unchanged. The visual predictor is parameter-frozen, but its tactile-conditioned forward pass remains differentiable with respect to the adapter input.

The implementation expresses this objective as a velocity-residual loss. Let v_0 be the frozen predictor's velocity without tactile augmentation and v_Z its velocity with augmentation. The predicted residual is $v_Z - v_0$, while the target residual is $(\varepsilon - y) - v_0$. Their difference is

$$(v_Z - v_0) - \big[(\varepsilon - y) - v_0\big] = v_Z - (\varepsilon - y). \tag{21}$$

Consequently, the main-text conditional flow objective also describes this implementation. The unaugmented baseline is evaluated without gradients; the augmented path propagates gradients to the tactile adapter and encoder. The squared error is spatially weighted, normalized by the weighted valid element count for each sample–horizon pair, and averaged over valid pairs. Reference ROI weights are 1 for the hand region and 0.1 for the background.

Future-pressure prediction. The pressure predictor conditions on detached cached visual context, the tactile context, and a detached upcoming window of edited actions. It therefore regularizes the shared tactile encoder without sending gradients through the action condition into the residual readout. Its reference architecture uses four Transformer decoder layers with width 256 and eight attention heads.

The prediction target is pressure rather than a learned tactile embedding. A fixed codec canonicalizes hand orientation and computes

$$y^r_u = \frac{P^{+,r}_u - \mu^r_u}{\max(\sigma^r_u, 10^{-3})}, \tag{22}$$

using training-set statistics in the same spatial convention. Each hand has 441 pressure sites. The future-hand validity and spatial sensor masks determine which sites contribute to the flow loss. The pressure objective averages over all valid sites, rather than averaging two per-hand losses with equal weight. In the contact-filtered variant, its future-hand mask also includes the gate computed from current history; it does not use a gate computed from the future target.

C.3 History Construction and Cached Training

Visual history burn-in. A training sample begins with a fresh context bank. Earlier observations are processed without gradients to populate it, and the final observation is processed with gradients.

The current output context remains attached to both action and visual-prediction objectives. The reference sampler uses at most nine visual observations per sequence, subject to available history. This procedure provides history-aware supervision without unbounded backpropagation through the bank.

Tactile initialization and frozen-context optimization. Contrastive pretraining operates on the normalized tactile and pose representations before tactile reference subtraction, as defined in the main text. The pose encoder is not part of the deployed policy. Cached editor training loads the tactile encoder and optimizes it jointly with the residual editor, tactile conditioning adapter, and pressure predictor. The visual branch and visual predictor remain frozen. Both cached visual context and base-action conditions are detached; the losses in this training path do not update the visual branch.

Event-driven execution. A visual update refreshes the base action chunk and visual context. Between visual updates, each tactile event encodes the current observed window and computes a correction for the active chunk's editable positions. Invalid or gated-out hands keep their base actions. A new visual update replaces the active chunk and its associated context. Both future-state predictors are absent from this execution path. Configured observation and command rates are scheduling parameters, not measurements of inference latency.

D Action and Evaluation Conventions

D.1 Action Layout and Temporal Alignment

Action coordinates. The hand-motion interface uses $H = 16$ action positions and 192 channels per position. Of these, 138 channels contain motion data and 54 are padding. Table 4 gives the component groups used by the losses. Padding is excluded from supervision and from valid sampled action components; it is not an additional action degree of freedom.

Table 4: Hand-motion action layout. Indices are zero-based, half-open intervals. Joint channels store 21 three-dimensional local positions per hand.

Component	Left hand	Right hand	Channels per hand
Wrist translation increment	$[0, 3)$	$[69, 72)$	3
Wrist Euler rotation increment	$[3, 6)$	$[72, 75)$	3
Palm-local joint coordinates	$[6, 69)$	$[75, 138)$	63
Padding	$[138, 192)$		—

Chunk positions and auxiliary action windows. The visual branch and tactile editor operate at 39.7 Hz and 119.1 Hz, respectively. Each visual update produces a new base action chunk. Between visual updates, the tactile editor uses incoming tactile observations to revise the unexecuted actions in the active chunk. The elapsed time $\Delta t_{t,\tau}$ is measured from the start of that chunk rather than inferred from the tactile update index.

For future-pressure conditioning, the action window begins at $k_0 = \lceil f_{\mathrm{cmd}} \Delta t_{t,\tau} \rceil$, where f_{cmd} is the action command rate, and contains the next eight edited actions. A floating-point tolerance is applied at integer boundaries. Positions beyond the chunk are zero-padded and marked invalid. This auxiliary window-indexing rule is distinct from the loss horizon and from the temporal mask that marks already executed actions as non-editable.

D.2 Evaluation Masks and Aggregation

Physical hand-motion errors. The hand-motion evaluator supports a declared horizon H_{eval} of eight or sixteen positions. A hand is eligible only if its complete evaluated prefix is valid; a truncated prefix is not silently substituted for the declared horizon. Predictions and targets are compared in the physical action coordinates used by the evaluator.

Table 5: Visual-only future hand-motion prediction. GigaHands uses its five official activity families; OakInk2 and TACO use their complete held-out test sets. The GigaHands average gives equal weight to the five families. Lower prediction error is better. Gray highlights T-WLA. **Bold**/underline indicate the best/second-best distinct reported errors within each metric and column, including ties. Frequency is reported in Hz and is not ranked. Dashes indicate unavailable results and are excluded from ranking.

	GigaHands						OakInk2	TACO	
Method	Cook.	Office	Craft.	Entert.	Housew.	Avg.	Overall	Overall	Freq.
Wrist translation error (cm, ↓)									
VITRA	1.16	1.25	1.11	1.44	1.37	1.27	0.53	0.97	19.4
SmolVLA	0.84	0.81	0.73	1.01	0.98	0.87	0.50	**0.72**	23.0
MemoryVLA	1.32	1.26	1.23	1.35	1.35	1.30	1.05	1.20	8.8
T-WLA	**0.63**	**0.60**	**0.56**	**0.69**	**0.74**	**0.65**	**0.44**	0.75	39.7
Wrist rotation geodesic error (degrees, ↓)									
VITRA	5.32	5.79	5.13	6.72	7.00	5.99	2.03	4.76	19.4
SmolVLA	3.75	3.91	3.69	5.07	5.23	4.33	2.03	3.68	23.0
MemoryVLA	6.52	6.54	6.07	6.97	7.25	6.67	5.27	6.66	8.8
T-WLA	**3.19**	**3.41**	**3.13**	**4.01**	**4.45**	**3.64**	**1.38**	**3.40**	39.7
Wrist-relative finger-joint MPJPE (cm, ↓)									
VITRA	1.48	1.32	1.36	1.41	1.36	1.39	2.04	2.74	19.4
SmolVLA	0.96	0.94	0.94	0.95	0.95	0.95	1.13	1.03	23.0
MemoryVLA	4.00	3.91	3.78	3.90	3.99	3.92	3.92	4.02	8.8
T-WLA	**0.43**	**0.43**	**0.41**	**0.44**	**0.45**	**0.43**	**0.37**	**0.46**	39.7

Wrist translation increments are accumulated within the prediction window before computing the Euclidean error, reported in centimeters. Euler rotation increments are converted using the xyz convention and composed by left multiplication, $R_k = \delta R_k R_{k-1}$ with $R_0 = I$ before the first increment. The reported rotational error is

$$e_{\mathrm{rot}}(R, \widehat{R}) = \frac{180}{\pi} \arccos\left(\mathrm{clip}_{[-1,1]} \frac{\mathrm{tr}(\widehat{R}^\top R) - 1}{2} \right). \tag{23}$$

Finger-joint MPJPE compares the stored palm-local positions directly, averaging joints 1–20 and reporting centimeters. Joint 0 is the fixed local origin and is excluded; the evaluator does not subtract a predicted wrist position from already local joint coordinates.

Sequence-level aggregation. For each metric, errors are first summed over eligible hand–time entries within each sequence and divided by their valid count. Sequences with no eligible entries are excluded. Environment-level means then weight these sequence means equally. The EgoTouch macro-average gives equal weight to its five environment means; the Tachin overall score gives equal weight to its eligible sequence means. This aggregation differs from pooling all frames across a dataset and should not be replaced by a sample-weighted mean from a different evaluation entry point.

Contact-conditioned evaluation. In the contact-filtered protocol, a sample is admitted when at least one supervised hand satisfies the causal gate in Eq. 16. Per-hand masks then exclude noncontact hands from action and future-pressure supervision. These exclusions define a contact-conditioned subset of the held-out data, not the complete held-out test set. Retaining the base action for an excluded hand does not insert that hand into the metric denominator. Comparisons under this protocol require a common sample set, hand mask, and prediction horizon; results on an unfiltered test set represent a separate evaluation protocol.

E Visual-Only Action Prediction

The visual-only experiment tests whether the architecture retains strong visual prediction capability when touch is unavailable. Tactile observations and tactile supervision are absent during both training and evaluation: only RGB, language, pose history, visual recurrent memory, and the visual

action expert are active. Consequently,

$$a = a^{\text{base}}, \qquad \mathcal{L}_{\text{visual-only}} = \mathcal{L}_{\text{act}} + \lambda_{\text{vis}}\mathcal{L}_{\text{vis}}. \tag{24}$$

This setting is therefore not a cross-modal transfer experiment; it directly evaluates the standalone quality of the visual branch.

Table 5 shows that T-WLA remains competitive without touch and achieves the best completed result on eight of the nine dataset-level metric comparisons. On GigaHands, T-WLA obtains average translation, rotation, and joint errors of 0.65 cm, $3.64°$, and 0.43 cm. Relative to SmolVLA, these values correspond to reductions of 25.3%, 15.9%, and 54.7%, respectively. The improvement is consistent across all five activity families rather than being concentrated in a single category.

On OakInk2, T-WLA reaches 0.44 cm translation error, $1.38°$ rotation error, and 0.37 cm joint MPJPE, reducing the corresponding SmolVLA errors by 12.0%, 32.0%, and 67.3%. On TACO, T-WLA achieves the lowest completed rotation and joint errors at $3.40°$ and 0.46 cm, improving over SmolVLA by 7.6% and 55.3%. Its translation error is 0.75 cm, slightly above SmolVLA's 0.72 cm but still 22.7% below VITRA. These results show that separating tactile correction from visual action generation does not weaken the visual predictor; instead, the visual memory and predictive supervision provide a strong action prior even when the tactile branch is absent.

F Real-Robot Setup and Demonstration Collection

Hardware. Our platform is a dual-arm Linker A7black robot with a 7-DoF arm and a Linker Hand L6 dexterous hand on each side. The hands provide fingertip tactile sensing. We mounted an Intel RealSense D435i on a custom 3D-printed bracket to capture an egocentric view of the workspace and calibrated its pose relative to the robot.



(a) Rrobot hardware platform (b) Data glove

Figure 6: **Robot platform and data glove**

Teleoperation. An operator wears Wuji data gloves to capture finger articulation and HTC VIVE Tracker 3.0 devices at the wrists to track wrist motion. Two SteamVR Base Station 2.0 units track the VIVE devices. The tracked wrist poses and glove measurements are mapped to the robot arms and dexterous hands to collect task demonstrations; no VR headset is required.

Demonstration budget. We collect 30 teleoperated episodes for each of the four real-robot tasks: egg handling, single-sheet pickup, paper-cup nesting, and paper-bag opening.

G Additional Qualitative Results

We provide additional qualitative comparisons on EgoTouch and Tachin to complement the examples shown in Fig. 4.

Throughout this section, VITRA, SmolVLA, MemoryVLA, and T-Rex refer to their tactile-augmented variants used in Table 1, whereas T-WLA uses its native tactile-conditioned interface.

We follow the same visualization protocol as Fig. 4. For each example, the source RGB observation is shown at t_0, while the ground-truth pose and all model predictions correspond to $t_0 + 100\,\mathrm{ms}$. Camera-frame overlays visualize the complete predicted hand pose, whereas wrist-aligned views emphasize differences in finger articulation.

The value below each prediction is the per-example wrist-aligned MPJPE over the 20 non-root hand joints at $t_0 + 100\,\mathrm{ms}$. These values are intended only as diagnostics for the displayed examples and should not be interpreted as dataset-level test averages. They isolate finger-articulation error after wrist alignment and are therefore distinct from the wrist-translation and wrist-rotation metrics reported in Table 1.

Across the displayed examples, T-WLA consistently produces fingertip configurations that more closely match the ground truth and achieves the lowest per-example wrist-aligned MPJPE among the compared methods. This mirrors the quantitative results in Table 1: the largest gains from T-WLA occur in finger-joint MPJPE, substantially exceeding the relative improvements in wrist translation and rotation. Together, these results suggest that T-WLA is particularly effective at predicting fine-grained fingertip and finger-articulation dynamics.

G.1 Additional EgoTouch Qualitative Examples

Figures 7 and 8 show six additional EgoTouch examples. We first visualize predictions in the camera frame and then use wrist-aligned renderings to isolate differences in finger articulation.

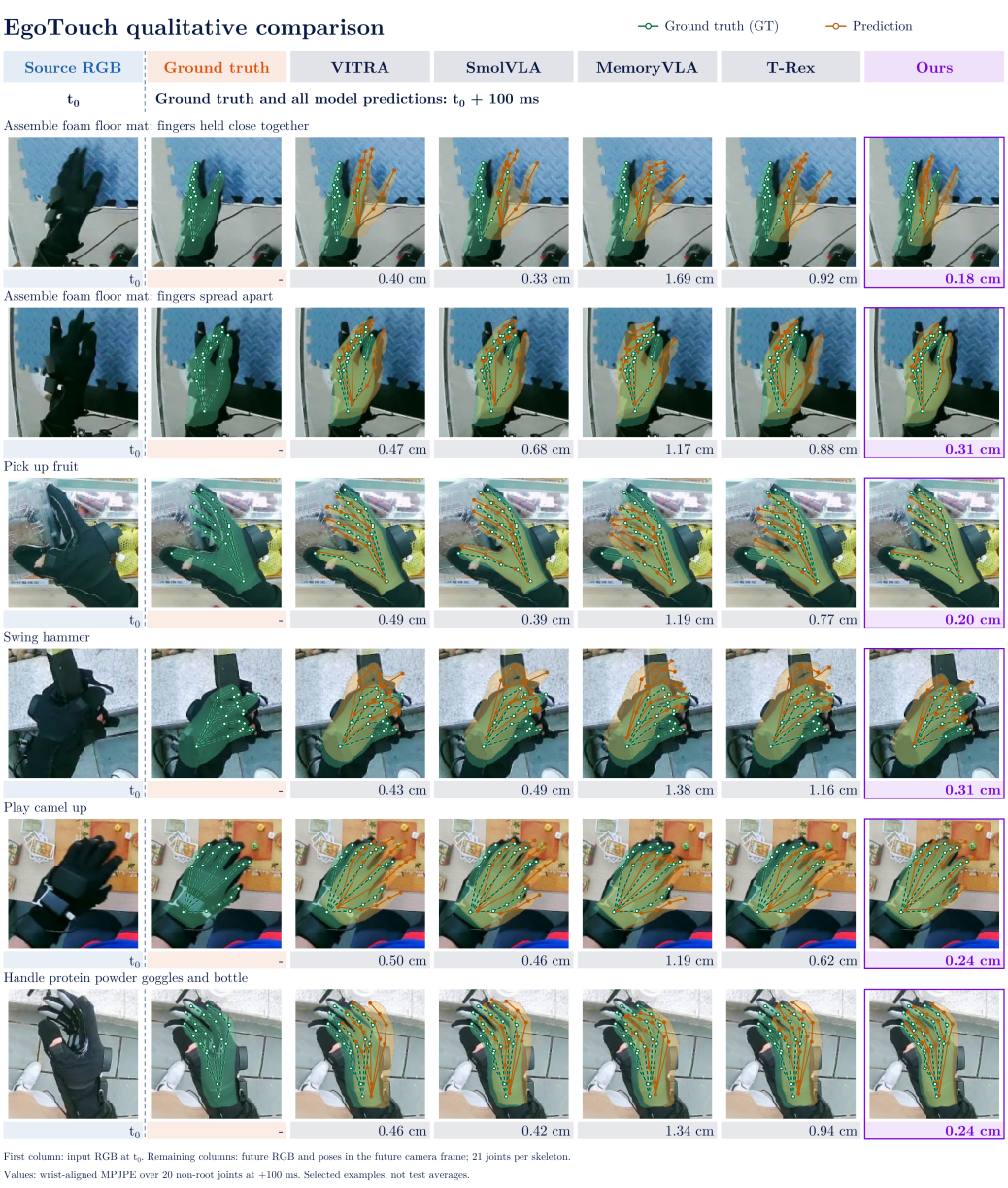


Figure 7: **Additional camera-frame qualitative comparisons on EgoTouch.** Each row shows the source RGB observation at t_0, followed by the ground-truth hand pose and predictions from VITRA, SmolVLA, MemoryVLA, T-Rex, and T-WLA at $t_0 + 100\,\text{ms}$. Ground truth is shown in green and model predictions in orange. Values below each prediction report the corresponding wrist-aligned finger-joint MPJPE in centimeters.



Figure 8: **Additional wrist-aligned fingertip comparisons on EgoTouch.** The same six examples from Fig. 7 are shown after wrist alignment to emphasize finger articulation independently of global wrist motion. Within each example, all hands share the same alignment, scale, and $+60°$ visualization rotation. The ground-truth hand mesh is shown in green, with ground-truth fingertips marked by green circles and predicted fingertips marked by orange crosses. Values below each prediction report wrist-aligned MPJPE over the 20 non-root hand joints at $t_0 + 100\,\text{ms}$. These values are per-example diagnostics rather than dataset-level test averages.

G.2 Additional Tachin Qualitative Examples

Figures 9 and 10 show six additional Tachin examples covering diverse hand orientations and finger configurations. We first visualize the future hand-pose predictions directly in the camera frame, where the source RGB observation is shown at t_0 and the ground-truth and predicted poses correspond to $t_0 + 100\,\text{ms}$. This view preserves both global wrist motion and finger articulation, allowing direct comparison of the complete predicted hand pose.

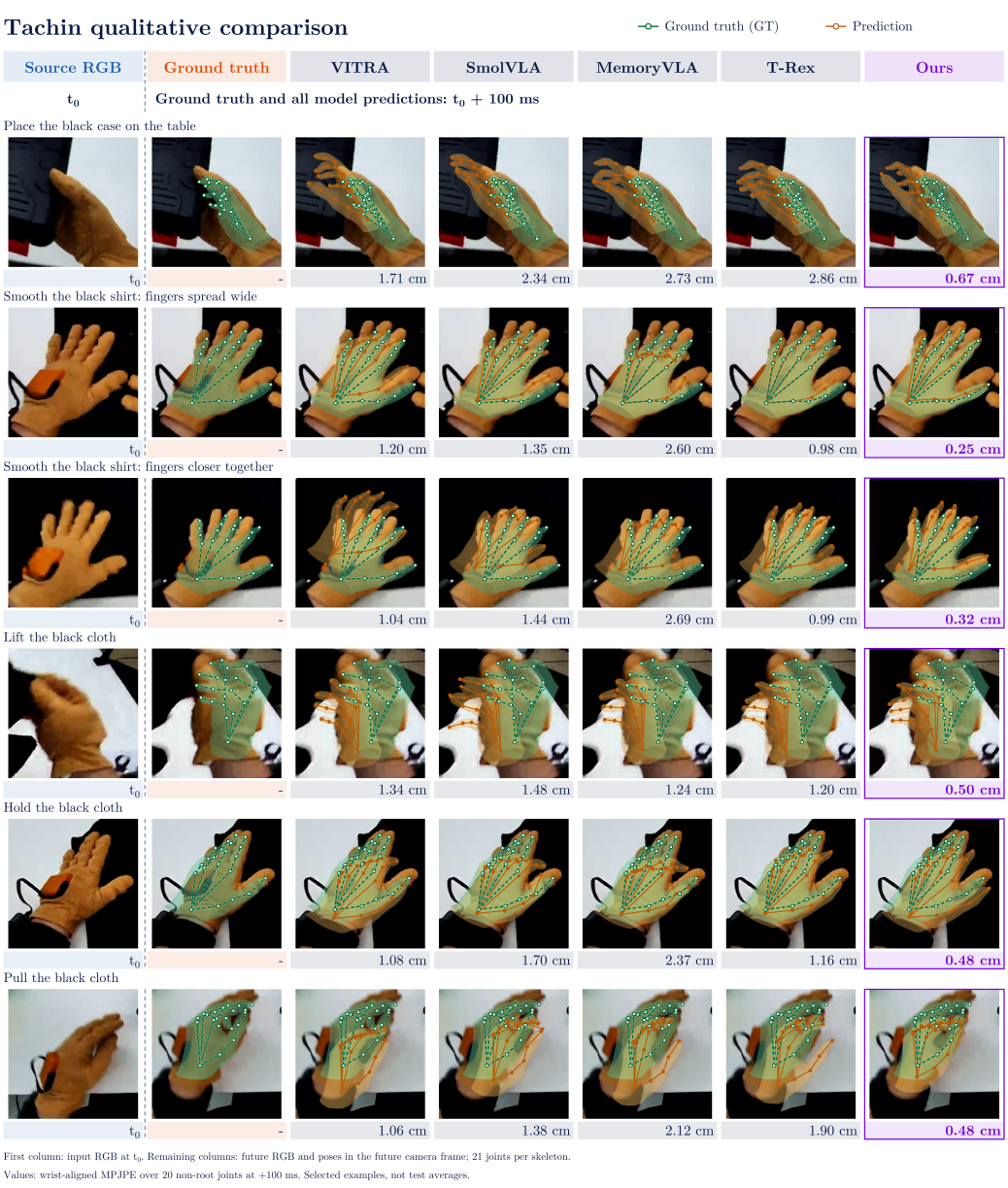


Figure 9: **Additional camera-frame qualitative comparisons on Tachin.** Each row shows one source RGB observation at t_0, followed by the ground-truth hand pose and predictions from VITRA, SmolVLA, MemoryVLA, T-Rex, and T-WLA at $t_0 + 100\,\text{ms}$. Ground-truth hand poses are shown in green and model predictions in orange. This camera-frame visualization preserves both global wrist motion and finger articulation. The values below each prediction report the corresponding wrist-aligned finger-joint MPJPE in centimeters.

We then show wrist-aligned fingertip-detail renderings for the same examples. Removing global wrist displacement makes differences in finger articulation easier to inspect independently of the hand's position in the camera frame. Across the displayed examples, T-WLA more closely matches the ground-truth fingertip configuration and achieves the lowest per-example wrist-aligned MPJPE in every row. These qualitative examples complement the dataset-level quantitative results reported in Table 1.

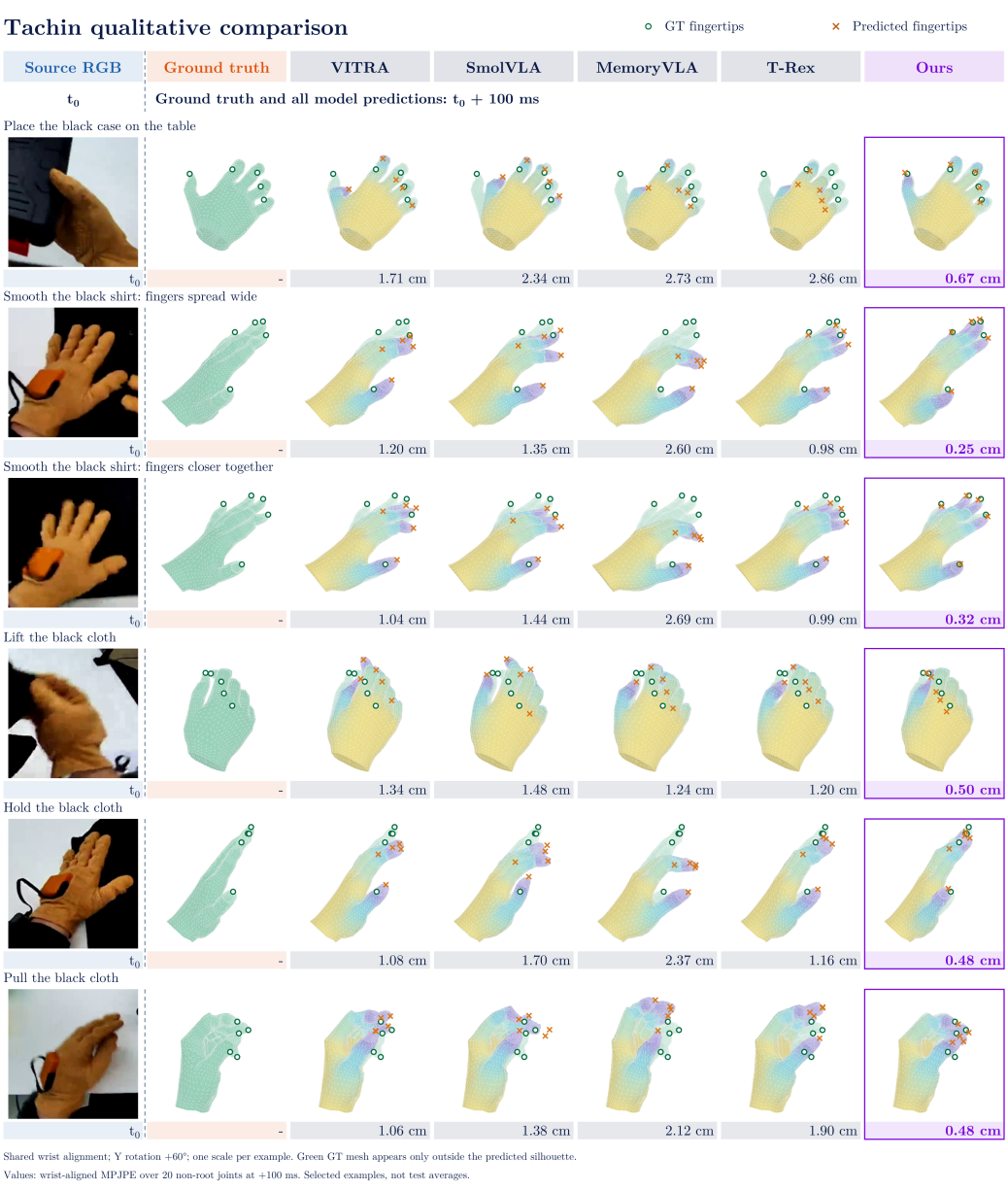


Figure 10: **Additional wrist-aligned fingertip comparisons on Tachin.** The same six examples from Fig. 9 are shown after wrist alignment to emphasize finger articulation independently of global wrist motion. Within each example, all hands share the same alignment, scale, and $+60°$ visualization rotation. The ground-truth hand mesh is shown in green, with ground-truth fingertips marked by green circles and predicted fingertips marked by orange crosses. Values below each prediction report wrist-aligned MPJPE over the 20 non-root hand joints at $t_0 + 100\,\mathrm{ms}$. These values are per-example diagnostics rather than dataset-level test averages.