EXEC-VTLA: EXECUTION-AWARE VISION-TACTILE-LANGUAGE-ACTION WITH MEASURED MOTION AND CONTACT HISTORIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language-action (VLA) models have demonstrated strong capabilities in robotic manipulation, but the effects of an action occur at the contact interface, which often lies in the blind spot of visual observation. Without sufficient information about actual motion and contact changes, policies may continue motions that are no longer appropriate after contact, leading to manipulation failures. To address this limitation, we present Exec-VTLA, an execution-aware vision-tactile-language-action policy that conditions future actions on measured motion and contact histories. Its core is a multi-source action head that uses future action queries. Each future action query jointly reads four complementary conditions through shared attention: visual-language context for task semantics, robot state for the current configuration, measured state transitions for the motion that actually occurred, and bilateral temporal-domain tactile observations for changes at the contact interface. The action head jointly predicts an action chunk relative to a common reference state, incorporating recent execution information into the generation of the next action segment. On the UniVTAC benchmark, Exec-VTLA achieves an average success rate of 68.17% across six contact-rich tasks. It further achieves an average success rate of 80.00% across four tasks on a physical robot platform. Ablation studies support the effectiveness of measured motion history, tactile input, and the proposed multi-source fusion.

1 INTRODUCTION

Vision-language-action (VLA) models can use pretrained vision-language knowledge to generate continuous robot control and have demonstrated strong transfer across multi-task policy learning (Brohan et al., 2023; Zitkovich et al., 2023; Kim et al., 2025; Ghosh et al., 2024; Black et al., 2025b). However, Contact-rich manipulation requires local physical information that is not reliably available from vision. The contact regions can be occluded by the object or the end-effector, and slip, collision, local deformation, and contact stability may not be identifiable from RGB (Fig. 1(a)). Tactile sensors such as GelSight and DIGIT provide access to local contact information for fine manipulation (Yuan et al., 2017; Lambeta et al., 2020), but how to effectively use such information to generate the next action segment remains a challenge for VLA policies.

Recent tactile policies and VTLA methods have advanced tactile representation, generalization, multimodal fusion, and feedback for robotic manipulation (Yuan et al., 2026; Zhao et al., 2025; Huang et al., 2025; Bi et al., 2026; Zhang et al., 2026; Zhou et al., 2026; Wang et al., 2026a). However, tactile feedback alone provides an incomplete account of execution, as contact changes do not explicitly describe how much the robot actually moved. A remaining challenge is how to jointly use measured motion and contact changes to guide the next action segment. The current robot state describes the configuration at the time of observation, whereas measured state transitions explicitly encode the recent motion leading to that configuration. For example, in Put Bottle (Fig. 1), visual and language cues can guide the robot toward the target, but after contact, RGB observations may not reliably reveal whether the bottle is seated, tilted, or still sliding. Tactile feedback provides local contact information that complements vision, while measured motion history indicates the robot's actual displacement during the interaction. Together, these signals provide evidence about both the evolving contact state and the extent to which motion is being realized, helping the policy determine

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.

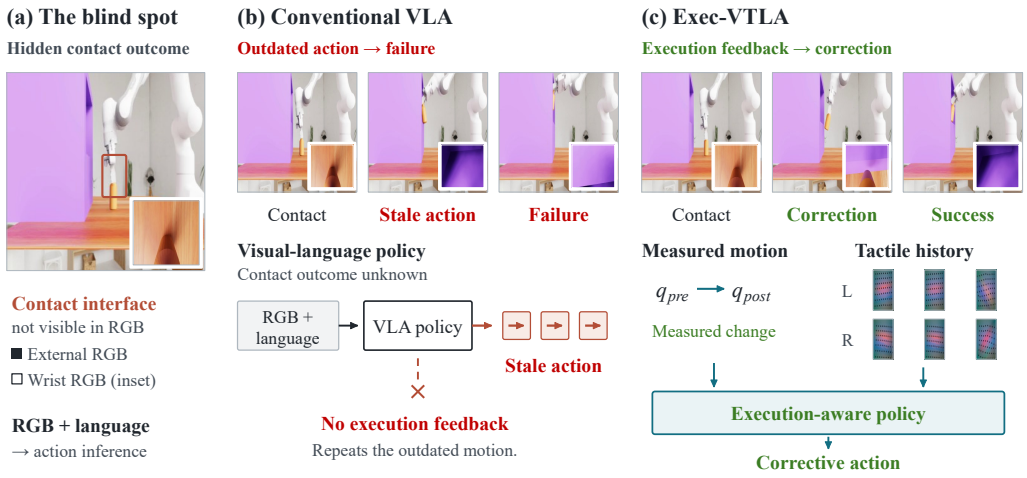


Figure 1: Problem motivation and execution-aware correction. (a) The local contact outcome is not reliably observable from external or wrist RGB views. (b) Without execution feedback, a conventional VLA repeats a stale motion after contact and the rollout fails. (c) Exec-VTLA conditions the next action on measured motion history and bilateral tactile history; the illustrated rollout corrects the motion and succeeds. Insets show wrist-camera images; L and R denote the two tactile streams.

whether to continue or adjust the next action segment. If this execution information is not adequately incorporated into action generation, the policy may continue a motion that is no longer appropriate (Fig. 1(a)). We refer to this failure mode as the action-contact outcome gap. The illustrated rollouts show an unsuccessful execution in which the policy continues an inappropriate motion (Fig. 1(b)) and a successful execution in which motion is adjusted using execution feedback (Fig. 1(c)).

We present Exec-VTLA, an execution-aware vision-tactile-language-action policy that conditions action generation on measured motion and contact histories. Measured motion history is computed from pre- and post-execution robot states and captures how much the robot actually moved, while contact history is represented by bilateral temporal tactile observations and captures changes at the contact interface. Exec-VTLA integrates these histories with visual-language task context and the current robot state in an action head built around future action queries. These execution signals enter the action decoder as dedicated condition tokens, preserving access to motion and contact information during action generation. Each query reads the relevant conditions through shared attention, and the action head jointly predicts an action chunk relative to a common reference state, using recent motion and contact observations to guide the next action segment. The common reference state gives all predicted actions a consistent geometric interpretation, while interactions among future action queries allow the decoder to model dependencies within the chunk.

Our contributions are as follows.

- We present Exec-VTLA, an execution-aware policy that incorporates complementary motion and contact histories into action generation. Measured state transitions capture the robot's actual motion, while bilateral temporal tactile observations capture changes at the contact interface.
- We design a future-action-query decoder that jointly integrates these execution histories with task context and robot state. Shared attention allows each future action query to access the conditioning information and other action representations, enabling joint prediction of a short action chunk relative to a common reference state.
- We evaluate Exec-VTLA on the UniVTAC benchmark and on an AgileX NERO arm with an embedded Tac3D visuotactile sensor. Task-level comparisons, ablations of execution inputs and decoder fusion, and execution-sequence analysis assess the policy's performance and the roles of its key components in contact-rich manipulation.

2 Related Work

2.1 Vision-Language-Action Policies

Action-chunk and generative architectures laid the foundations for modern VLAs. ACT (Zhao et al., 2023) predicts temporally coherent action chunks, Diffusion Policy (Chi et al., 2025) models multi-modal action sequences through iterative denoising, and Flow Matching (Lipman et al., 2023) provides an efficient paradigm for continuous action generation. Generalist policies were then scaled across tasks, embodiments, and environments. RT-1 (Brohan et al., 2023) established real-world transformer control, RT-2 (Zitkovich et al., 2023) connected internet-pretrained representations, RT-X (O'Neill et al., 2024) and Octo (Ghosh et al., 2024) scaled across heterogeneous robot data, and OpenVLA (Kim et al., 2025), π_0 (Black et al., 2025b), $\pi_{0.5}$ (Black et al., 2025a), RDT-1B (Liu et al., 2025), SmolVLA (Shukor et al., 2025), and GR00T N1 (NVIDIA et al., 2025) diversified along action generation and deployment. On the decoding side, the VLA-Adapter and S^2-VLA (Wang et al., 2026b; Xie et al., 2026) represent a direction that replaces large pretrained backbones with lightweight policy modules and bridge attention, while Audio-VLA (Wei et al., 2025) adds contact audio as a non-visual observation channel. Across these systems, observations remain centered on cameras, language, and robot state (Vaswani et al., 2017; Bai et al., 2025). While vision-language-action policies show strong transfer performance, contact and slip are visually unobservable during execution, and contact outcomes from executed actions lie outside the visual field.

2.2 Tactile Sensing and Tactile-Conditioned Manipulation

Optical tactile sensors have established high-resolution contact sensing. GelSight (Yuan et al., 2017) measures contact geometry and force through surface deformation, DIGIT (Lambeta et al., 2020) offers a compact low-cost design, and Taxim (Si & Yuan, 2022) provides a simulation model for GelSight sensors. Tactile representation and policy pretraining methods improve tactile perception or downstream control. VITaL (George et al., 2025) learns visuo-tactile pretraining representations, UniVTAC (Chen et al., 2026) provides data synthesis and benchmarking in simulation, and FTP-1 (Yuan et al., 2026) and N_0-VTLA (NeoteAI Team & Fudan TEAI Team, 2026) pretrain generalist tactile policies and VTLA foundation models on large-scale tactile data, yet they focus on representation learning and transfer across sensors rather than on how contact outcomes condition specific future actions. Existing tactile policy and VTLA methods demonstrate the value of tactile feedback for contact-rich manipulation. Tactile-VLA (Huang et al., 2025) unlocks tactile generalization in VLAs, VLA-Touch (Bi et al., 2026) enhances VLAs with dual-level tactile feedback, TacVLA (Zhang et al., 2026) improves occluded and contact-rich tasks through contact-aware fusion, Tac-Forcing (Zhou et al., 2026) resolves the temporal mismatch between tactile acquisition and execution via streaming action generation, and ViTaR (Wang et al., 2026a) casts touch as bounded residual modulation atop a frozen VLA, while classical force control and visuo-tactile grasping (Khatib, 1987; Calandra et al., 2018) established the value of local physical feedback. These methods are typically tied to specific sensors, tasks, or fusion designs. Touch enters the policy as additional tokens, feedback modules, or residuals, but the outcome of contact of an executed action is not aligned with the future action positions that must respond to it, leaving the action-contact outcome gap unaddressed. Exec-VTLA closes this gap by making policy execution-aware: it directly conditions each future action row inside the action head on the measured motion history and the contact history encoded with the transferable T3 representation (Zhao et al., 2025), so recent execution information enters the generation of the next action segment.

3 Method

Exec-VTLA is built around the action-contact outcome gap. In contact-rich manipulation, the contact interface is often hidden from RGB, so a conventional vision-language policy may continue an inappropriate motion after contact. The two real rollouts in Fig. 1 illustrate an unsuccessful execution without effective correction and a successful execution in which contact and measured motion feedback guide subsequent actions. Our goal is to establish an action-outcome feedback loop in which the policy uses recent contact observations and measured motion history to generate each new action segment.

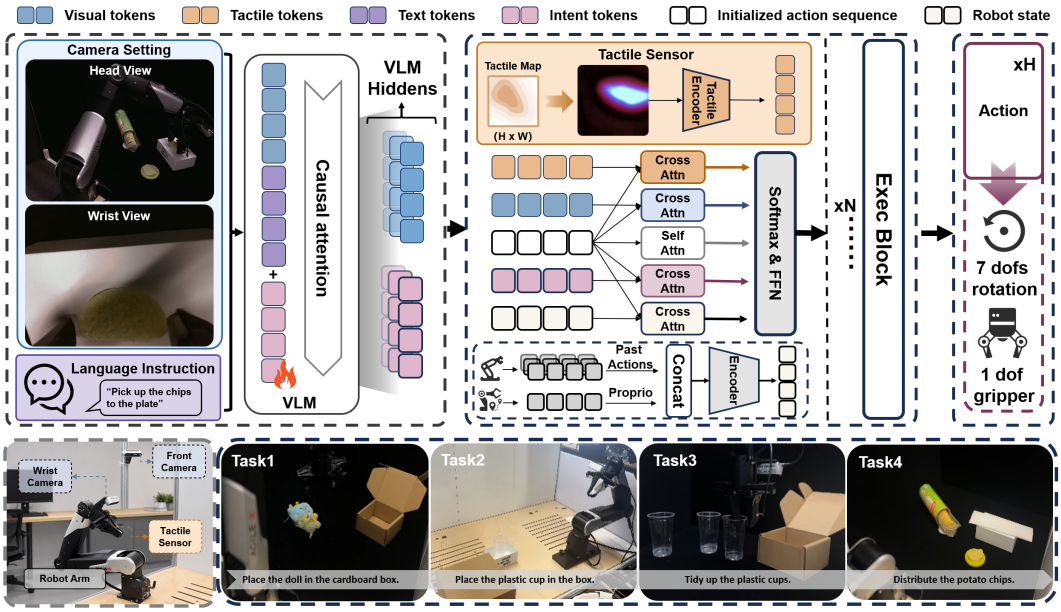


Figure 2: Overview of Exec-VTLA. Multi-view RGB observations and the language instruction are encoded by the vision-language backbone, while bilateral tactile observations, current proprioception, and measured motion history enter the action head as separate condition streams. The query-centered action head integrates these sources through shared multi-source attention and decodes the next action chunk. Updated motion and tactile histories condition subsequent predictions. The bottom row shows the physical platform and four evaluation tasks.

Fig. 2 provides an overview of Exec-VTLA. A vision-language backbone processes camera observations, language instructions, and learnable action tokens to produce task-context and action-query features. In parallel, a tactile encoder extracts features from bilateral temporal tactile observations, while measured motion history and the current robot state are encoded as additional conditions. Starting from an initialized action sequence, the action decoder jointly integrates these conditions with the backbone features through shared attention, allowing each future action representation to access both external conditions and other action representations. Stacked decoding blocks jointly predict a short action chunk relative to a common reference state, specifying the translation, rotation, and gripper commands. Updated tactile observations and measured state transitions are then incorporated into subsequent policy inputs to guide the next action segment. The lower part of the figure shows the physical robot setup and four real-world evaluation tasks. We first formalize the problem, then present the action decoder, and close with training and inference.

3.1 Problem Formulation

Following the standard vision-language-action (VLA) formulation, a policy maps the observation available at decision time t to a horizon-H future action chunk,

$$A_{t:t+H-1} = \pi_{\text{VLA}}(V_t,\, L_t,\, P_t), \tag{1}$$

where π_{VLA} denotes a VLA policy, $A_{t:t+H-1} = \{a_t^i\}_{i=0}^{H-1}$ is the future action sequence, H is the action horizon, V_t denotes the current multi-view RGB observations, L_t is the language instruction, and P_t is the current proprioceptive robot state. This formulation conditions action generation primarily on the currently available observation and does not explicitly represent the contact changes or actual motion of already executed actions. After contact, the policy may therefore continue an outdated motion without knowing how the previous action segment changed the contact state.

Exec-VTLA augments the policy input with tactile contact history and measured motion history,

$$A_{t:t+H-1} = \pi_{\text{Exec-VTLA}}(V_t,\, L_t,\, P_t \mid T_t,\, U_t), \tag{2}$$

where T_t is the constructed bilateral temporal tactile condition and U_t is the state condition that combines the current proprioceptive state with measured motion history. Each target action is defined relative to the robot state at the current decision time,

$$a_t^i = q_{t+i+1} - q_t, \qquad i = 0, \ldots, H-1, \tag{3}$$

where q_t denotes the robot state at time t. $A_{t:t+H-1}$ represents a short-horizon future trajectory relative to the reference state q_t, rather than a sequence of accumulated action increments.

3.2 EXEC-VTLA

A pretrained vision-language model processes the current multi-view observation V_t and language instruction L_t into visual tokens and text tokens. The action head additionally receives the tactile condition T_t and state condition U_t, allowing action generation to jointly use task context, current robot configuration, measured motion, and contact changes.

Temporal tactile conditioning. The real-world setup uses one Tac3D tactile sensor and a history length K. Let G_{t-r} denote the tactile image at relative time index r. Each frame is encoded by the pretrained T3 encoder and projected to the action-head dimension. We write the resulting temporal tactile condition compactly as

$$T_t = \left\{ \mathrm{Proj}_T\big(E(G_{t-r})\big) + e_r^{\mathrm{time}} + e^{\mathrm{sensor}} \right\}_{r=0}^{K-1}, \tag{4}$$

where e_r^{time} and e^{sensor} are learnable embeddings. The implementation uses validity masks for unavailable or padded frames and supplies the resulting tactile condition to every Exec Block.

State and executed-action history. The current proprioceptive state is projected to the action-head dimension. The recent executed-action history is encoded by a learned history encoder and augmented with learned position embeddings. The resulting state condition is

$$C_t^{\mathrm{state}} = \mathrm{Concat}\left(\mathrm{Proj}_P(P_t), \left\{ \mathrm{Proj}_U(u_k) + e_k^{\mathrm{hist}} \right\}_{k=t-M}^{t-1} \right), \tag{5}$$

where M is the configured history length and u_k denotes the k-th entry supplied by the executed-action-history field. This history is produced by the data preprocessing and deployment logging pipeline and is encoded by the action head through a learned history encoder. It is therefore distinct from the current state projection and is not recomputed from the command inside the action head. A history mask suppresses invalid padded entries. The projected current state is also added to the initial action-token embeddings through a state-to-query projection, allowing the action chunk to start from the current robot configuration.

Future-action-query-centered Exec Block. The action head initializes H learned action-token representations, with one token corresponding to each future action row. These tokens form the query states of every Exec Block. The action head uses five condition branches in the order used by the implementation that $\mathcal{S} = \{\mathrm{self}, \mathrm{intent}, \mathrm{visual}, \mathrm{state}, \mathrm{tactile}\}$. These sources correspond to the five token types in Fig. 2. self denotes the dynamically updated action tokens X. intent denotes the VLM hidden states at the inserted action-query positions. visual denotes visual tokens produced from camera observations. state denotes the robot-state tokens C_t and $\mathrm{tactile}$ denotes the bilateral temporal tactile tokens T_t. The self source is updated at every decoder block through its action-token attention path, allowing the action tokens to coordinate across future action positions.

For Exec Block b, let $X^{(b)} \in \mathbb{R}^{H \times d_h}$ be the current action-token representation. Let $H_a^{(b)}$ and $H_v^{(b)}$ denote the intent and visual hidden states from the VLM, and let C_t^{state} and T_t denote the state and tactile conditions. The query and source-specific key/value projections are

$$Q^{(b)} = X^{(b)} W_Q, \qquad K^{(b)} = \left[K_s^{(b)} \right]_{s \in \mathcal{S}}, \qquad V^{(b)} = \left[V_s^{(b)} \right]_{s \in \mathcal{S}}, \tag{6}$$

where the self branch is computed from $X^{(b)}$, the intent branch from $H_a^{(b)}$, the visual branch from $H_v^{(b)}$, the state branch from C_t^{state}, and the tactile branch from T_t. Each branch has independent key and value projections. The state-history and tactile masks are concatenated with the source scores before normalization.

All source scores are concatenated along the token dimension and normalized by one shared softmax. The action-token representation is updated as

$$X^{(b+1)} \leftarrow \mathrm{FFN}\left(X^{(b)} + W_o \,\mathrm{softmax}\left(\frac{Q^{(b)}K^{(b)\top}}{\sqrt{d_h}}\right)V^{(b)}\right), \tag{7}$$

The action head stacks N_{block} such blocks. Learned action tokens therefore query VLM action-query features, visual-task features, the current state, executed history, and temporal tactile changes, while the self branch coordinates the H future action rows.

Bounded action decoding. The final action-token representation is normalized and projected to the action space. An elementwise $\tanh$ bounds the normalized output

$$\widehat{A}_t = \tanh\left(W_{\mathrm{out}}\mathrm{LN}(X^{(N_{\mathrm{block}})}) + b_{\mathrm{out}}\right), \qquad \mathcal{L} = \frac{1}{H}\sum_{i=0}^{H-1}\left\|\widehat{a}_t^i - a_t^i\right\|_1. \tag{8}$$

For an action chunk length H and action dimension D_a, the predicted action tensor has shape $\mathbb{R}^{H\times D_a}$. Each row corresponds to one future action position, and all rows are trained against the demonstration action chunk with the mean L1 objective. Normalized predictions are subsequently denormalized prior to execution.

3.3 Training and Inference

The model is trained end-to-end, jointly full fine-tuning the vision-language backbone and the action head. The visuotactile encoder uses the transferable T3 representation (Zhao et al., 2025) and remains frozen. In inference, actions execute row by row with state-jump and safety checks. The measured pre- and post-states are written into the next inference history, forming an action–outcome feedback loop with the same semantics used during training.

4 Experiments

This section addresses three questions. First, does temporal tactile conditioning support stable closed-loop control on standardized contact-rich tasks? Second, do tactile history, executed-action history, and multi-source Exec Block attention each contribute to performance? Third, can the complete vision-tactile-action path close the loop on a physical platform with a single Tac3D sensor? We answer these questions with matched evaluation on six UniVTAC tasks, controlled component comparisons, and four real-world tasks on an AgileX NERO arm. Training and deployment share the same action representation, normalization, and safety protocol. We next describe the evaluation setup and results.

4.1 Experimental Setup

Evaluation is conducted on two platforms. UniVTAC provides standardized task definitions and execution protocols for comparing tactile manipulation policies, while the AgileX NERO arm with an embedded Tac3D visuotactile sensor tests whether the complete multimodal control path runs on physical sensors and actuators. Model details are provided in Section 3.

The hardware platform consists of two subsystems. The robotic manipulation subsystem is built upon an AgileX NERO 7-DoF arm, which provides the motion flexibility required for tabletop manipulation, with an AgileX gripper mounted at the end-effector for grasping and object interaction. The visuotactile perception subsystem redesigns the original gripper structure and embeds a Tac3D visuotactile sensor into the end-effector, allowing the sensor to directly observe the local interaction between the gripper and the manipulated object while preserving the original grasping functionality. The resulting platform forms a unified vision–tactile–action system that supports evaluation of the proposed VTLA policy.

The baselines cover three families: ACT (Zhao et al., 2023) as an action-chunk behavior-cloning baseline, $\pi_{0.5}$ (Black et al., 2025a) as a flow-matching generalist VLA, Tactile-VLA (Huang et al., 2025) and FTP-$\pi_{0.5}$ as tactile-augmented VLAs, and FTP-1 (Yuan et al., 2026) as the previously

Table 1: Comparison with SOTA methods on UniVTAC. Exec-VTLA and methods evaluated under our protocol use 100 evaluation seeds per task. Avg. is the unweighted mean over all six tasks. The best result in each column is marked with **value***, while the second-best result is shown in **value**.

Method	LiftBottle	PullKey	LiftCan	PutBtl	InsHole	InsTube	Avg.
ACT (Zhao et al., 2023)	42	28	20	28	19	45	30.33
VITaL* (George et al., 2025)	72	47	8	32	25	34	36.33
UniVTAC-ACT* (Chen et al., 2026)	71	46	29	31	25	56	43.00
$\pi_{0.5}$ (Black et al., 2025a)	**97***	38	**72***	16	31	41	49.17
Tactile-VLA (Huang et al., 2025)	**97***	32	15	10	41	56	41.83
FTP-$\pi_{0.5}$ (Yuan et al., 2026)	**77**	30	26	19	47	72	45.17
FTP-1 (Yuan et al., 2026)	**97***	**48**	65	**47**	**64**	**79**	**66.67**
Exec-VTLA (Ours)	62	**50***	**68**	**80***	**67***	**82***	**68.17***

Table 2: Ablation study of Exec-VTLA on UniVTAC. Each task uses 100 evaluation seeds, and Avg. is the unweighted mean over the six tasks. The variants remove tactile input or measured motion history, replace multi-source attention with feature concatenation, or reduce the action horizon to one step (H1). Bold indicates the highest value in each column.

Experiment setting	LiftBottle	PullKey	LiftCan	PutBottle	InsHole	InsTube	Avg.
w/o tactile	54	38	58	71	59	73	58.83
w/o measured action history	57	42	62	74	62	76	62.17
concat fusion	59	44	64	76	64	78	64.17
H1 single-step policy	48	28	51	60	50	57	49.00
Full Exec-VTLA	**62**	**50**	**68**	**80**	**67**	**82**	**68.17**

strongest cross-sensor tactile foundation policy. VITaL* and UniVTAC-ACT* retain the original benchmark statistics. All six UniVTAC tasks use 100 evaluation seeds, and each platform task uses 20 independent trials. A trial is successful only if the task finishes completely within the action budget without unsafe actions or runtime errors.

4.2 Main Results on UniVTAC

Table 1 addresses the first question. Exec-VTLA reaches 68.17% overall and 69.75% without the two Lift tasks, both the highest in the table. Relative to FTP-1, the previously strongest tactile baseline, it improves Pull-out Key, Put Bottle, Insert Hole, and Insert Tube by 2, 33, 3, and 3 points and improves Lift Can by 3 points, making it the only method that exceeds FTP-1 on all four insertion, pull-out, and placement tasks. On Lift Bottle, Exec-VTLA reaches 62%, below the 97% of FTP-1 and $\pi_{0.5}$, indicating that pure lifting relies more on existing policy priors than on local contact adjustment. Notably, $\pi_{0.5}$ (Black et al., 2025a) is strongest on the two Lift tasks but reaches only 16% on Put Bottle, while Tactile-VLA (Huang et al., 2025) averages 41.83%, showing that simply attaching tactile input to a VLA does not produce effective contact conditioning. Overall, the Exec-VTLA advantage concentrates on tasks that require continuous contact adjustment.

4.3 Ablation Studies

Table 2 addresses the second question by removing temporal tactile conditioning, executed-action history, and multi-source Exec Block attention, together with an H1 single-step action-policy comparison. The full Exec-VTLA reaches 68.17% on average, while removing tactile input reduces the average to 58.83% (-9.34 points), indicating that contact observations are important when the task state changes at the gripper-object interface. Removing executed-action history reduces performance to 62.17% (-6.00 points), showing that the current observation alone does not fully reveal how the previous action was executed. Replacing structured multi-source attention with concatenation reduces the average to 64.17% (-4.00 points), suggesting that organizing conditions across future action rows matters beyond modality availability, whereas H1 reaches 49.00%, demonstrating the importance of coordinated short-horizon action prediction. The task-wise pattern is consistent with these roles. Tactile removal most affects Pull-out Key, Lift Can, and Insert Tube, which con-

Table 3: Success rates on the AgileX NERO arm with an embedded Tac3D sensor. Each task uses 20 independent trials. The best result in each column is marked with **value***, while the second-best result is shown in **value**.

Method	Toy	Cup	Long-Tactile	Chips	Avg.
ACT (Zhao et al., 2023)	14/20 (70%)	8/20 (40%)	4/20 (20%)	8/20 (40%)	42.50%
Tactile-VLA (Huang et al., 2025)	16/20 (80%)	12/20 (60%)	8/20 (40%)	11/20 (55%)	58.75%
$\pi_{0.5}$ (Black et al., 2025a)	**18/20 (90%)***	***14/20 (70%)***	**12/20 (60%)***	***14/20 (70%)***	***72.50%***
Exec-VTLA (Ours)	**18/20 (90%)***	**18/20 (90%)***	**12/20 (60%)***	**16/20 (80%)***	**80.00%***

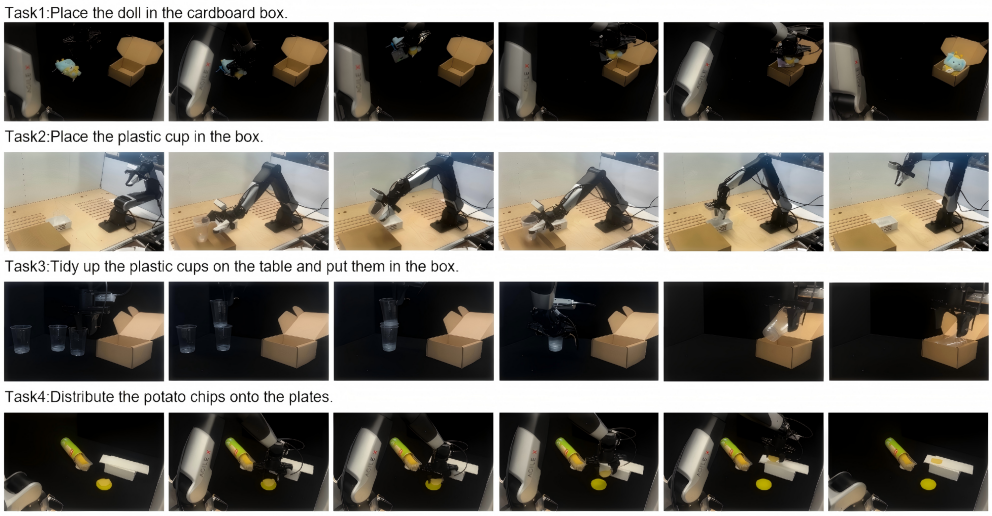


Figure 3: Real-world robotic manipulation experiments are conducted on the AgileX NERO platform with the Tac3D tactile sensing module. The figure illustrates complete execution sequences of four tasks, including chip grasping, cup grasping, toy grasping, and long-horizon manipulation.

tain changing or partially occluded contact states. Removing executed-action history most affects Pull-out Key and Put Bottle, where the next action depends on the displacement produced by the preceding action. Concatenation remains below the structured Exec Block on every task, and H1 shows the largest degradation on Pull-out Key, Put Bottle, and Insert Tube, which require multiple temporally coordinated corrections. Together, these results support the complementary roles of tactile sensing, execution history, structured fusion, and future-action decoding.

4.4 Evaluation on the AgileX NERO Arm with an Embedded Tac3D Sensor

The hardware platform (Fig. 3) consists of a robotic manipulation subsystem and a visuotactile perception subsystem. The manipulation subsystem is built upon an AgileX NERO 7-DoF arm with an AgileX gripper mounted at the end-effector, and the perception subsystem embeds a Tac3D visuotactile sensor into the redesigned gripper so that the sensor directly observes the local interaction between the gripper and the object while preserving the original grasping functionality. Table 3 addresses the third question. Each of the four platform tasks uses 20 independent trials. Entries report successful trials over total trials, followed by the success rate. Exec-VTLA reaches an 80.00% average, obtaining the best results in Chips, Cup, and Toy Grasping and matching $\pi_{0.5}$ (Black et al., 2025a) on the long-horizon tactile task. The largest gains over ACT (Zhao et al., 2023) and Tactile-VLA (Huang et al., 2025) occur on delicate- and rigid-object grasping, while the long-horizon tie indicates remaining planning and state-coverage limits.

4.5 Interpreting Execution-Aware Conditioning

We investigate whether method induces an execution-feedback chain from contact change, to motion correction, to convergence. To this end, we examine one successful **put_bottle_in_shelf** rollout and

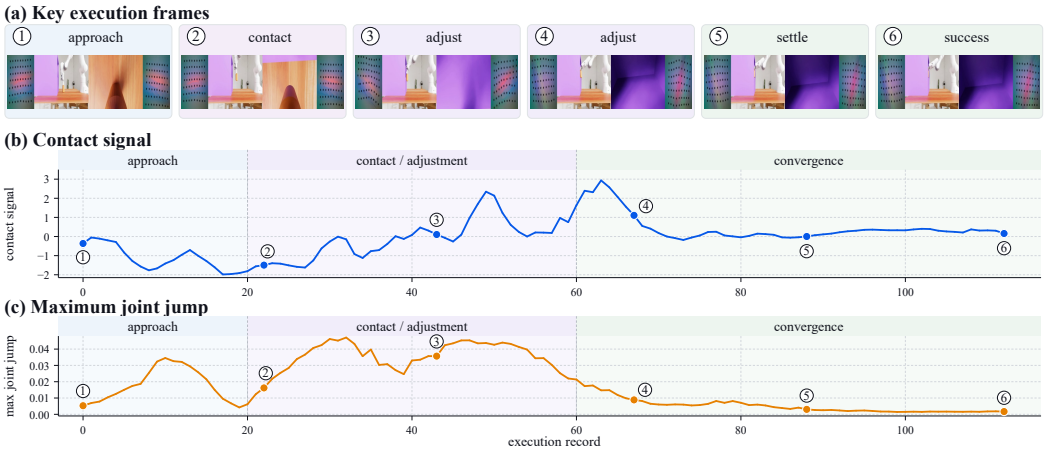


Figure 4: Execution signals during one successful Put Bottle rollout. (a) Six selected records show approach, contact, adjustment, settling, and completion; each record contains two RGB views and bilateral tactile observations. (b) Contact signal and (c) maximum joint jump over the same rollout. Numbered markers link the images to records 0, 22, 43, 67, 88, and 112, and the background marks the approach, contact/adjustment, and convergence phases. Joint jumps are largest during contact/adjustment and decay as the rollout converges.

align its executed frames, internal contact estimate, and maximum joint jump on a shared timeline (Fig. 4): panel (a) shows key execution frames throughout the manipulation process, panel (b) visualizes the scalar internal contact estimate c_t produced from sequential temporal tactile input, and panel (c) illustrates the maximum joint jump $j_t = \max_k |q_{t,k} - q_{t-1,k}|$ calculated between adjacent execution records. For quantitative correlation analysis, we adopt the first-order temporal change of the contact signal, defined as $\Delta c_t = c_t - c_{t-1}$.

We partition the entire manipulation rollout into six continuous execution-state stages, marked at records $0, 22, 43, 67, 88, 112$ (markers 1–6 in panel (a)). These key timestamps are selectively sampled at critical execution-state transitions rather than uniform time intervals, covering initial approach, contact formation, active correction, stabilization, settling, and final convergence. The three monitored quantities exhibit consistent phase-dependent coupling throughout the whole task process. Before physical contact occurs, the contact estimate remains stably near its baseline value (-0.36) and the robotic motion is minimal with negligible joint variation (≈ 0.005). During active correction after contact establishment, the tactile estimate fluctuates significantly between -1.81 and $+2.14$, the maximum joint jump peaks at 0.047, and the same-step correlation between Δc_t and j_t reaches $r = +0.50$ (N execution records), which gradually decreases to ≈ 0.39 after a three-step temporal shift. In the final convergence stage, the joint jump continuously decays from 0.038 to 0.0016, indicating stable task completion. The synchronized evolution of tactile variation and robotic motion constitutes solid mechanism-level evidence that tactile contact feedback directly drives real-time motion correction and eventual task convergence.

Complementing this detailed single-rollout mechanism analysis, the tactile ablation experiments in Table 2 further verify the functional importance of tactile conditioning. The removal of tactile sensory information reduces the task success rate from 68.17% to 58.83%, providing intuitive task-level evidence for the essential contribution of tactile feedback. Together, the cooperative execution-level and task-level experimental results strongly support that METHOD learns a closed-loop contact-correction-convergence execution chain rather than relying on a naive open-loop manipulation plan.

5 DISCUSSION

Exec-VTLA adapts actions via motion and tactile history to avoid redundant motions under varying contact and occlusion. Its unified representation ensures stable deployment, and future work will extend this feedback mechanism for long-horizon correction.

AI USE STATEMENT

We utilized generative AI tools, including codex gtp5.6sol, to assist with code writing and enhance the clarity and readability of the manuscript. All conceptual design, experimental planning, result analysis, and final verification of the work were completed by the authors. The authors take full responsibility for the final content of this paper.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, et al. Qwen3-VL technical report, 2025. URL https://arxiv.org/abs/2511.21631. arXiv:2511.21631.

Jianxin Bi, Kevin Yuchen Ma, Ce Hao, Mike Zheng Shou, and Harold Soh. VLA-Touch: Enhancing vision-language-action model with dual-level tactile feedback. *IEEE Robotics and Automation Letters*, 11(7):8487–8494, 2026. doi: 10.1109/LRA.2026.3692345.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, et al. $\pi_{0.5}$: 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. URL https://proceedings.mlr.press/v305/black25a.html.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, et al. π_0: A vision-language-action flow model for general robot control. In *Proceedings of Robotics: Science and Systems*, 2025b. doi: 10.15607/RSS.2025.XXI.010.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, et al. RT-1: Robotics transformer for real-world control at scale. In *Proceedings of Robotics: Science and Systems*, 2023. doi: 10.15607/RSS.2023.XIX.025.

Roberto Calandra, Andrew Owens, Dinesh Jayaraman, Justin Lin, Wenzhen Yuan, Jitendra Malik, Edward H. Adelson, and Sergey Levine. More than a feeling: Learning to grasp and regrasp using vision and touch. *IEEE Robotics and Automation Letters*, 3(4):3300–3307, 2018. doi: 10.1109/LRA.2018.2852779.

Baijun Chen, Weijie Wan, Tianxing Chen, Xianda Guo, Congsheng Xu, Yuanyang Qi, Haojie Zhang, Longyan Wu, Tianling Xu, Zixuan Li, Yizhe Wu, Rui Li, Xiaokang Yang, Ping Luo, Wei Sui, and Yao Mu. UniVTAC: A unified simulation platform for visuo-tactile manipulation data generation, learning, and benchmarking, 2026. URL https://arxiv.org/abs/2602.10093. arXiv:2602.10093.

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. doi: 10.1177/02783649241273668.

Abraham George, Selam Gano, Pranav Katragadda, and Amir Barati Farimani. VITaL pretraining: Visuo-tactile pretraining for tactile and non-tactile manipulation policies. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 258–264, 2025. doi: 10.1109/ICRA55743.2025.11128336.

Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Quan Vuong, Ted Xiao, Pannag R. Sanketi, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.090.

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, 2025. URL https://arxiv.org/abs/2507.09160. arXiv:2507.09160.

Oussama Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, 1987. doi: 10.1109/JRA.1987.1087068.

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. URL https://proceedings.mlr.press/v270/kim25c.html.

Mike Lambeta, Po-Wei Chou, Stephen Tian, Brian Yang, Benjamin Maloon, Victoria Rose Most, Dave Stroud, Raymond Santos, Ahmad Byagowi, Gregg Kammerer, Dinesh Jayaraman, and Roberto Calandra. DIGIT: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation. *IEEE Robotics and Automation Letters*, 5(3):3838–3845, 2020. doi: 10.1109/LRA.2020.2977257.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

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*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/49f80e4d2471ad4f2edf4f5f1ab62339-Abstract-Conference.html.

NeoteAI Team and Fudan TEAI Team. N_0-VTLA: Scaling vision-tactile-language-action model with latent tactile tokens, 2026. URL https://arxiv.org/abs/2607.23782. arXiv:2607.23782.

NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, et al. GR00T N1: An open foundation model for generalist humanoid robots, 2025. URL https://arxiv.org/abs/2503.14734. arXiv:2503.14734.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, et al. Open X-Embodiment: Robotic learning datasets and RT-X models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903, 2024. doi: 10.1109/ICRA57147.2024.10611477.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, Simon Alibert, Matthieu Cord, Thomas Wolf, and Remi Cadene. SmolVLA: A vision-language-action model for affordable and efficient robotics, 2025. URL https://arxiv.org/abs/2506.01844. arXiv:2506.01844.

Zilin Si and Wenzhen Yuan. Taxim: An example-based simulation model for GelSight tactile sensors. *IEEE Robotics and Automation Letters*, 7(2):2361–2368, 2022. doi: 10.1109/LRA.2022.3142412.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://papers.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Yi Wang, Renjun Wu, Jinyan Liu, and Xuesong Li. ViTaR: Visuo-tactile residual adaptation for foundation VLA manipulation, 2026a. URL https://arxiv.org/abs/2608.15816. arXiv:2608.15816.

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. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(22):18638–18646, 2026b. doi: 10.1609/aaai.v40i22.38931.

Xiangyi Wei, Haotian Zhang, Xinyi Cao, Siyu Xie, Weifeng Ge, Yang Li, and Changbo Wang. Audio-VLA: Adding contact audio perception to vision-language-action model for robotic manipulation, 2025. URL https://arxiv.org/abs/2511.09958. arXiv:2511.09958.

Zhipeng Xie, Zongyi Han, Xiangyi Wei, Shiliang Sun, Yang Li, and Jing Zhao. S^2-VLA: State-space guided vision-language-action models for long-horizon manipulation. In *Proceedings of the Thirty-Fifth International Joint Conference on Artificial Intelligence*, pp. 6245–6253, 2026. doi: 10.24963/ijcai.2026/695.

Chengbo Yuan, Zicheng Zhang, Mingjie Zhou, Wendi Chen, Yi Wang, Zhuoyang Liu, Dantong Niu, Shuo Wang, Hui Zhang, Wenkang Zhang, Yingdong Hu, Yuanqing Gong, Wanli Xing, Chuan Wen, Cewu Lu, Kaifeng Zhang, and Yang Gao. FTP-1: A generalist foundation tactile policy across tactile sensors for contact-rich manipulation, 2026. URL https://arxiv.org/abs/2606.13102. arXiv:2606.13102.

Wenzhen Yuan, Siyuan Dong, and Edward H. Adelson. GelSight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12):2762, 2017. doi: 10.3390/s17122762.

Kaidi Zhang, Heng Zhang, Zhengtong Xu, Zhiyuan Zhang, Md Rakibul Islam Prince, Xiang Li, Xiaojing Han, Yuhao Zhou, Arash Ajoudani, and Yu She. TacVLA: Contact-aware tactile fusion for robust vision-language-action manipulation, 2026. URL https://arxiv.org/abs/2603.12665. arXiv:2603.12665.

Jialiang Zhao, Yuxiang Ma, Lirui Wang, and Edward Adelson. Transferable tactile transformers for representation learning across diverse sensors and tasks. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 3766–3779. PMLR, 2025. URL https://proceedings.mlr.press/v270/zhao25c.html.

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.

Jianbo Zhou, Boyuan Zhao, Yuzheng Zhang, Yiyang Chen, Wenxin Chen, Qiuyue Li, Xiangyang Gu, Yuhan Cao, Xiao Xia, Yanzhe Hu, and Zhijie Deng. TacForcing: Streaming action generation with execution-time tactile feedback, 2026. URL https://arxiv.org/abs/2608.25798. arXiv:2608.25798.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183. PMLR, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

A Setup Details of Simulation Benchmarks

We evaluate Exec-VTLA on the six contact-rich tasks of the UniVTAC benchmark, including Lift Bottle, Pull-out Key, Lift Can, Put Bottle, Insert Hole, and Insert Tube. The tasks cover lifting, pulling, placement, and insertion interactions in which the relevant state can change after contact. Lift Bottle, Lift Can, and Put Bottle require a stable grasp followed by a placement whose outcome depends on how the object settles in the gripper. Pull-out Key requires overcoming initial resistance, so a policy that repeats the approach motion after contact cannot recover. Insert Hole and Insert Tube are the most precise tasks, since a small initial offset must be corrected through contact feedback rather than visual alignment alone.

All methods use the same task definitions, action representation, execution frequency, safety checks, and success criteria. Exec-VTLA is evaluated with 100 seeds per task, and the average is the arithmetic mean of the six task success rates. For every seed, the evaluator fixes the task instruction



Figure 5: Representative visual-tactile rollouts for six simulated manipulation tasks. Each row corresponds to the instruction printed above it, and time proceeds from left to right. At each selected step, the tactile observation is shown to the left of the external RGB view.

and the initial-condition generation rule, and records the complete action sequence, the measured pre-state and post-state pair of each executed action row, the bilateral tactile stream, and the final task state. A trial is successful only when the task is completed within the action budget without an unsafe action or runtime error. Failed trials are labeled by termination cause, including unsafe action, runtime error, early stop, and max-action termination, which are used for the failure analysis in the main paper.

B Setup Details of Real-World Evaluation

B.1 Hardware System

The real-world evaluation uses an AgileX NERO arm equipped with a Tac3D visuotactile sensor integrated into the end-effector. The arm has seven revolute joints and a parallel gripper, giving the eight-dimensional proprioceptive state and action vectors used throughout the paper. The platform provides three complementary observation channels: external RGB cameras for the scene view, joint encoders for proprioception, and bilateral tactile sensing at the gripper. The Tac3D sensor directly observes local changes at the gripper-object interface, complementing the external RGB views that are frequently occluded by the gripper or the manipulated object during contact. This combination mirrors the simulation setup, so the same observation interface, action representation, and history construction are shared between simulation training and real-world deployment.

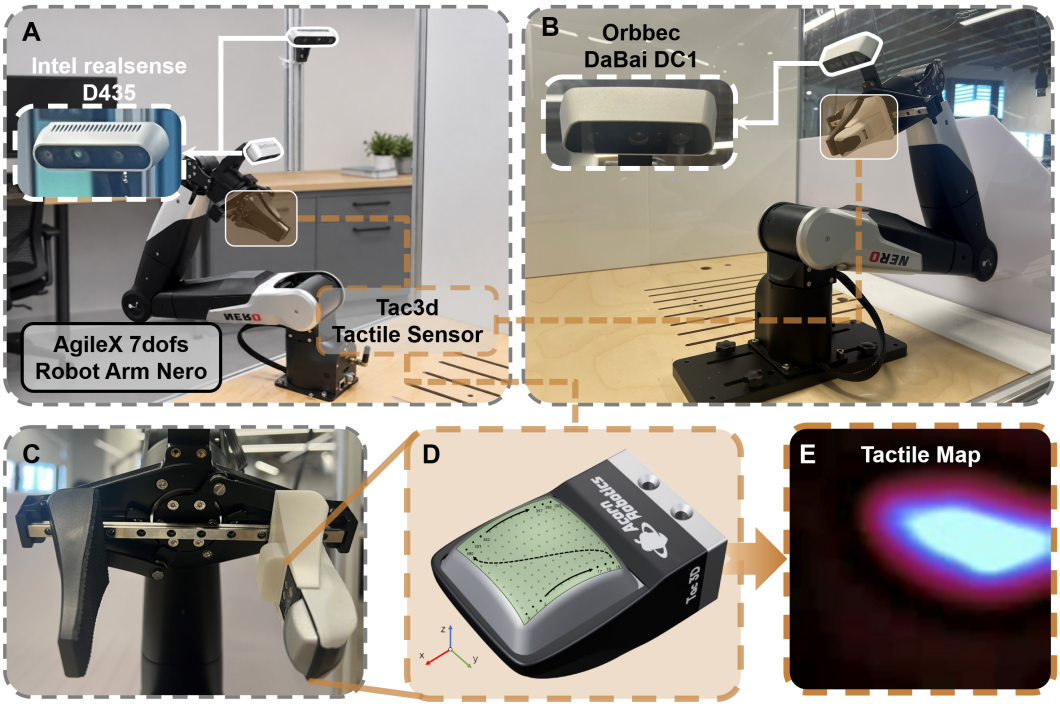


Figure 6: Real-world evaluation platform and sensing setup. (A, B) AgileX NERO 7-DoF arm with the Intel RealSense D435 and Orbbec DaBai DC1 camera configurations, respectively. (C) Close-up of the gripper with the integrated Tac3D visuotactile sensor. (D) Tac3D sensor and (E) an example tactile map.

B.2 Task Definitions and Evaluation Protocol

The platform evaluation contains four tasks: chip grasping, cup grasping, doll grasping, and long-horizon tactile manipulation. Chip grasping uses a fragile, low-rigidity object whose contact response is dominated by local deformation, so a stable grasp requires gentle, contact-aware closing rather than a fixed gripper command. Cup grasping uses a rigid object and mainly evaluates spatial grounding together with stable placement. Doll grasping uses a soft, irregular object whose shape cannot be predicted from RGB alone, making tactile feedback the primary source of grasp-state information. The long-horizon task chains multiple contact-dependent stages, so errors from earlier stages propagate unless the policy corrects them through execution feedback.

Each task uses 20 independent trials, and the platform score is the arithmetic mean of the four task success rates. Each trial starts from a randomized object configuration and runs under the same action budget and safety checks as simulation. For every trial, we record external RGB observations, robot proprioception, bilateral tactile observations, issued actions, measured pre-state and post-state pairs, and the final termination reason. The success criterion matches the UniVTAC criterion, so simulation and real-world results are directly comparable.

C Details of Training and Inference

C.1 Dataset Scale

Simulation training uses 100 demonstrations for each of the six UniVTAC tasks, totaling 600 episodes. Real-world training data contain 80 demonstrations for chip grasping, 80 for cup grasping, 80 for doll grasping, and 140 for long-horizon tactile manipulation, totaling 380 episodes. The long-horizon task receives more demonstrations because a single successful episode covers multiple contact-dependent stages and therefore provides a larger share of the correction-critical states.

Algorithm 1 Inference Process of Exec-VTLA

Input: Initial observation V_t, instruction L_t, state s_t, tactile history T_t, execution history U_t, budget B
Output: Termination status and execution record
for $t = 0, \ldots, T$ **do**
  Step 1: Multi-Source Encoding
    $(F_t^{\text{visual}}, I_t^{\text{intent}}) = \text{VLM}(V_t, L_t)$
    $C_t^{\text{state}} = \text{EncodeState}(s_t, U_t)$
    $C_t^{\text{tactile}} = \text{EncodeTactile}(T_t)$
  Step 2: Action Chunk Prediction
    $A_t = \text{ExecBlock}(I_t^{\text{intent}}, F_t^{\text{visual}}, C_t^{\text{state}}, C_t^{\text{tactile}})$
  Step 3: Row-Wise Execution
  for each action row $a_t^i \in A_t$ **do**
      $q^{\text{pre}} = \text{ReadRobotState}()$, execute a_t^i
      $q^{\text{post}} = \text{ReadRobotState}()$
    Step 4: Outcome Write-Back
      $U_t = \text{AppendAndTrim}(U_t, q^{\text{post}} - q^{\text{pre}})$
      $T_t = \text{UpdateTactileHistory}(\text{ReadBilateralTactile}())$
      $s_t = q^{\text{post}}$, $V_t = \text{ReadVisualObservation}()$
    if $\text{UnsafeJump}(a_t^i) \vee \text{RuntimeError}() \vee \text{TaskSuccess}()$ **then**
        terminate with the corresponding status
    end if
  end for
end for

Each demonstration stores multi-view RGB frames, proprioceptive states, bilateral tactile frames, and the corresponding eight-dimensional action targets, so the same data format serves both the vision-language backbone and the condition streams of the action head.

C.2 Training Objective

For each training sample, the target is an action chunk defined relative to the common anchor state. The action prediction loss is the mean L1 distance over the H future rows

$$\mathcal{L}_{\text{action}} = \frac{1}{H} \sum_{i=0}^{H-1} \left\| \widehat{a}_t^i - a_t^i \right\|_1 . \tag{9}$$

All H rows of one chunk are supervised jointly, so every future row, not only the first, remains accurate when the chunk is executed sequentially. The L1 objective is preferred over squared error because contact-rich demonstrations contain occasional large deviations at contact transitions, and the absolute-distance loss prevents these transitions from dominating the gradient. Since every target row is defined relative to the common anchor state, the supervised quantity and the measured execution feedback share the same geometric semantics, which keeps training and deployment consistent.

The vision-language backbone and the action head are jointly optimized. The visuotactile encoder uses the transferable T3 representation of (Zhao et al., 2025) and remains frozen unless an ablation states otherwise. Freezing the tactile encoder preserves the contact representation learned during its pretraining and reduces the number of trainable parameters, while the projection layers, temporal and side embeddings, action queries, and all Exec Block parameters are optimized end-to-end.

C.3 Inference Procedure

The inference procedure of Exec-VTLA operates as a temporally recurrent loop of multi-source conditioning and sequential decision-making, as detailed in Algorithm 1. This process requires only 10 GB of VRAM and can be deployed on the NVIDIA RTX 4060 Ti GPU.

The key advantage of this inference procedure lies in the outcome write-back in Step 4, which feeds the measured transition and the updated tactile history into the next Exec Block forward pass. One

Table 4: Architecture and training configuration of Exec-VTLA. Batch size is reported per device. The remaining entries specify the model dimensions, history lengths, optimization settings, and compute used in the reported experiments.

Setting	**Value**
Architecture	
Backbone	Qwen3-VL-2B-Instruct
VLM hidden dimension (d_h)	2048
Attention backend	SDPA
Gradient checkpointing	Enabled
Mixed precision	bf16
Tactile encoder	T3 (frozen)
Tactile image size	224×224
Action model	Exec Block
Action prediction	L1 regression
Exec Block hidden dimension	512
Action-token latent dimension	512
Exec Block depth (N_{block})	24
Attention heads per block	8
Dropout	0.0
Action queries and future action rows (H)	8
Action dimension (D_a)	8
Proprioceptive state dimension (D_s)	8
Executed-history length (M)	4
Training	
Optimizer	AdamW
Adam betas	$(0.9, 0.95)$
Adam epsilon	10^{-8}
Weight decay	10^{-6}
Gradient accumulation	1
Maximum gradient norm	1.0
Scheduler	Cosine with minimum learning rate
Minimum learning rate	10^{-7}
Base learning rate	5×10^{-7}
VLM interface learning rate	5×10^{-7}
Action model learning rate	5×10^{-5}
State-to-action learning rate	5×10^{-5}
Tactile-to-action learning rate	2×10^{-5}
Tactile time embedding rate	2×10^{-5}
Tactile sensor-identity embedding rate	2×10^{-5}
Warm-up steps	10
Training epochs	1
Max training steps	50,000
Batch size per device	16
Device	NVIDIA H100 GPU $\times$ 8
Random seed	42

forward pass produces a complete chunk of eight future action rows that share the same condition history, and the rows are then executed sequentially without regeneration inside the current chunk. Both histories are kept at fixed lengths, four measured transitions for the execution history and the most recent four frames from each sensing side for the tactile history, so the condition tensors have a constant shape across steps. Because every write-back uses measured states rather than commanded actions, the loop reacts to what the robot actually did, including blocked, slipped, or truncated motions, and the entire procedure runs within 10 GB of VRAM on a single consumer-grade GPU.

C.4 Training and Architecture Configurations

Table 4 summarizes the architecture and training configuration. Training uses single-node eight-GPU distributed training on NVIDIA H100 GPUs with Accelerate and bf16 mixed precision. The per-device batch size is 16, giving an effective batch size of 128. We adopt a layer-wise learning-

rate strategy: the VLM interface uses a small learning rate to preserve the pretrained vision-language representation, while the action head, state projection, and tactile projection use larger learning rates to accelerate policy alignment. The tactile time and side embeddings receive their own learning rates so that the identity encodings can adapt without disturbing the frozen contact features. Gradient clipping and a short warm-up stage stabilize the early phase of joint optimization. Frozen modules include the pretrained tactile module, the feature projection modules, and the direct VLM-to-state and VLM-to-tactile paths.

The vision-language backbone produces the visual-token and intent-token features that enter the action head. The action head is implemented as a stack of 24 Exec Blocks, each performing multi-source attention over five condition branches, namely self, intent token, visual token, robot state, and tactile features, under one shared softmax normalization. The tactile branch encodes the most recent four frames from each side with the frozen T3 encoder, followed by temporal and side identity embeddings. The state branch combines the current eight-dimensional proprioceptive state with the four most recent measured transitions. For action chunking, we adopt a chunk size of H, with one action query per future action row, and the final rows are decoded through layer normalization and a $\tanh$ bounded projection, yielding an 8×8 bounded action chunk.