VT-Bridge: Bridging Pretrained Foundation VLAs to VTLAs via Lightweight Residual Adaptation

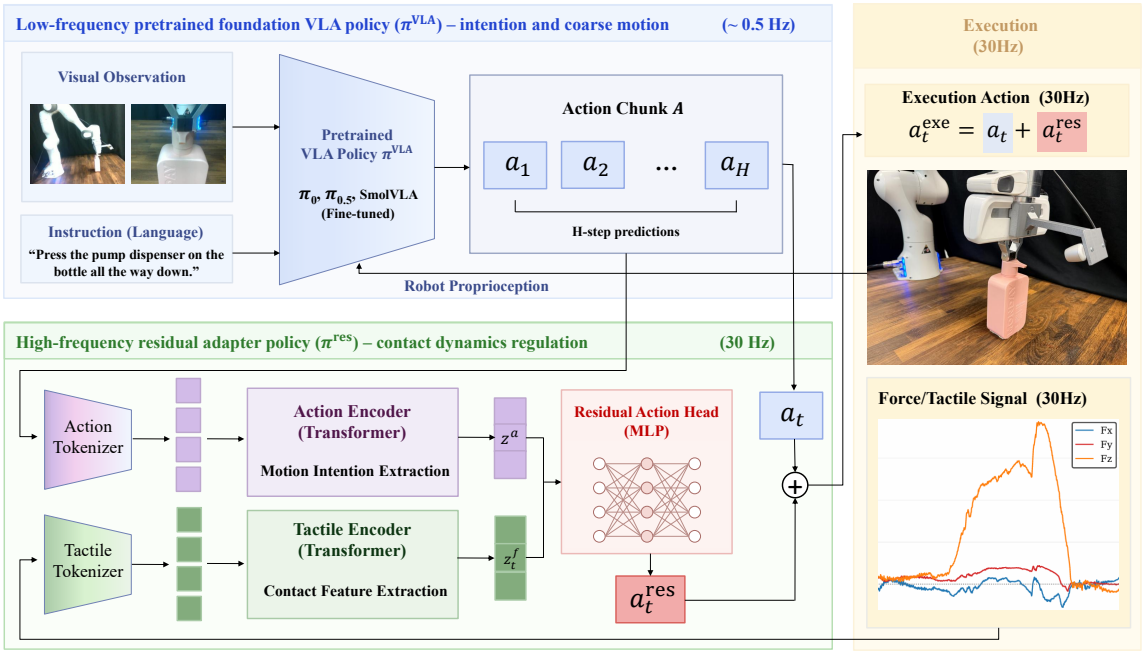

Fig. 1: Overview of VT-Bridge, a resource-efficient approach for bridging pretrained VLAs to VTLAs. It refines VLA-predicted actions using tactile feedback at the robot execution frequency. The residual-adapter architecture is applicable to different VLA backbones, with only its weights trained separately for each backbone.

***Abstract*—Vision-Tactile-Language-Action (VTLA) models have demonstrated clear advantages over Vision-Language-Action (VLA) models in contact-rich manipulation. However, developing VTLA models is severely constrained by the massive amounts of vision-tactile data and computational resources required. To address this bottleneck, we propose VT-Bridge, a lightweight residual adaptation strategy that bridges pretrained foundation VLAs to VTLAs. Rather than training a VTLA model from scratch or modifying the original architecture of a pretrained VLA, VT-Bridge employs an identical lightweight residual-adapter architecture across VLA backbones and uses backbone-specific weights to refine actions at the robot execution frequency. This design substantially lowers the data and training barriers. Specifically, it requires up to 50 vision-tactile demonstrations per task to fine-tune a VLA backbone and train a 0.98M-parameter residual adapter. Experiments with three representative VLA backbones (π_0, $\pi_{0.5}$, and SmolVLA) across four contact-rich manipulation tasks further demonstrate its consistent effectiveness across VLA architectures. On average, VT-Bridge raises the task completion rate from 11.7% with task-level VLA fine-tuning alone to 62.9%. Together, these findings demonstrate the broad applicability, effectiveness, and accessibility of VT-Bridge for contact-rich manipulation. The project page is available at this URL.**

Keywords: Vision-Language-Action Models, Vision-Tactile-Language-Action Models, Contact-Rich Manipulation

I. Introduction

Vision-Language-Action (VLA) models illuminate a viable pathway towards general-purpose robot intelligence [1], where a single pretrained policy can interpret natural-language instructions and execute diverse tasks across manipulation scenarios. Leveraging large-scale pretraining, VLA models show promising capabilities in tackling various tasks either zero-shot or through fine-tuning on a limited number of demonstrations [2]–[15]. Yet, even after task-specific fine-tuning, current VLAs continue to exhibit low success rates in real-world contact-rich manipulation, indicating that relying solely on non-contact modalities (vision, language, and robot kinematic state) for action prediction is insufficient for reliable physical interaction.

To address this limitation, recent studies have begun incorporating tactile feedback into VLA architectures, giving rise to Vision-Tactile-Language-Action (VTLA) models [16]–[34]. These efforts span varied fusion strategies and model architectures, yielding promising gains in contact-rich manipulation tasks. Moreover, some empirical ablation studies have isolated tactile contributions, underscoring the critical role of tactile sensing in reliable physical interaction [21]–[23].

However, developing VTLA models remains costly, with substantial resource demands in both data collection and model training. From the data perspective, paired vision-tactile trajectories are far scarcer than vision-only data in public robot datasets [22], [35]. Moreover, collecting high-quality tactile demonstrations requires fine force regulation, significantly elevating teleoperation complexity [36], [37].

From the model perspective, tactile encoders and cross-modal fusion modules introduce additional parameters and make the architecture more complex. These additions can also increase training costs. Constrained by these data and model bottlenecks, existing VTLAs commonly cover a significantly narrower range of tasks than general-purpose VLAs like π_0 [2]. This underscores a central question: *Can we adapt existing open-source VLAs into VTLAs at a manageable cost?*

To address this question, we propose VT-Bridge, a lightweight residual adaptation strategy that bridges pretrained foundation VLAs to tactile-aware VTLAs for contact-rich manipulation. As shown in Fig. 1, VT-Bridge introduces a lightweight residual adapter without modifying the pretrained VLA architecture. Based on the latest tactile feedback, the adapter refines each VLA-predicted action immediately before execution, thereby regulating contact dynamics at run-time. This design retains the general knowledge acquired through large-scale VLA pretraining, while enabling the tactile responsiveness required for contact-rich manipulation. Moreover, VT-Bridge supports various pretrained VLA backbones using an identical residual adapter architecture, with the adapter weights tailored to each backbone. It significantly reduces the resources required to build VTLA policies for downstream contact-rich tasks.

VT-Bridge possesses several key advantages:

- **Pretrained VLAs Reuse and Compatibility:** Instead of training a monolithic VTLA model from scratch, VT-Bridge preserves the pretrained VLA model and augments a pretrained VLA with a lightweight, tactile-conditioned residual adapter. This design retains the vision-language grounding and motion-generation capabilities acquired during large-scale VLA pretraining. The same adapter architecture can be applied to different VLA backbones by tailoring its parameters to each backbone, as demonstrated with π_0 [2], $\pi_{0.5}$ [3], and SmolVLA [38].
- **Low Data and Training Barriers:** Compared with training a VTLA model from scratch, VT-Bridge substantially lowers both data and training barriers. Specifically, it requires up to 50 vision-tactile demonstrations per task to fine-tune a VLA backbone and train a lightweight residual adapter with only 0.98M parameters. No additional large-scale training resources beyond those required for VLA fine-tuning are needed.
- **Run-time Reactivity:** As illustrated in Fig. 1, VT-Bridge adopts a slow–fast inference scheme that performs semantic task reasoning and tactile adaptation at different frequencies. The VLA operates at a lower frequency to interpret complex scenes and generate action chunks encoding task intent and coarse motion, while the lightweight residual adapter responds to contact changes at the execution rate.
- **Substantial and Consistent Performance Gains:** Despite its lightweight design and limited data requirements, VT-Bridge raises the average task completion rate from 11.7% under task-level VLA fine-tuning to 62.9%. The improvements are consistently observed across four real-world contact-rich manipulation tasks and three representative VLA backbones.

II. Related Work

A. Vision-Language-Action Models.

VLA models [4], [6], [8], [9], [12], [15] leverage large-scale vision-language pretraining to learn end-to-end robot policies that map visual observations and language instructions to low-level actions. Recent advances have evolved along several complementary architectural directions. Autoregressive models [8], [9], [13] formulate robot control as next-token prediction over discretized action tokens, naturally inheriting the training paradigm of vision-language models. Diffusion-based [5], [7], [10], [12], [14] approaches instead model continuous action distributions, producing smoother and more expressive behaviors at the cost of iterative inference. Flow-based methods, exemplified by π_0 [2], replace diffusion sampling with conditional flow matching to significantly improve inference efficiency, while dual-system architectures such as $\pi_{0.5}$ [3] and GR00T-N1 [11] decouple high-level semantic reasoning from low-level action generation for long-horizon manipulation. Despite these advances, existing VLA models primarily emphasize semantic reasoning and visual understanding. Their robustness in contact-rich manipulation remains limited, motivating the development of tactile-aware VLA models.

B. Vision-Tactile-Language-Action Models

To remedy this gap, VTLA models have emerged as a promising research direction. As illustrated in Fig. 2, existing VTLA methods can be broadly categorized according to how tactile information is fused:

- *Unified VTL Fusion:* As the most intuitive approach (Fig. 2(a)), methods in this category independently encode vision, touch, and language and feed the resulting representations into a unified multimodal transformer [19], [27], [29], [30], [33], [34].
- *Sequential VL-T Fusion:* Visual and language inputs are first jointly encoded, after which tactile information is incorporated at different stages. Tactile features can be fused through a Mixture of Experts (MoE) before action prediction (Fig. 2(b)) [22], [23], injected into the action decoder or action expert during action generation (Fig. 2(c)) [17], [18], [20], [21], [24], or used to refine previously predicted actions (Fig. 2(d) and Fig. 2(e)) [16], [26].
- *Others:* Tactile information can be fused at multiple stages [20], converted into continuous soft prompts for VLA reasoning [32], or used by a VLM to predict stiffness for compliant action execution [28].

Despite their promising performance on demonstrated contact-rich tasks, developing VTLA models remains costly due to the substantial vision-tactile data and computational resources required. VLA-Touch [26] takes an initial step toward adapting a pretrained VLA into a VTLA (Fig. 2(d)), but the broader potential of this adaptation paradigm remains largely underexplored.

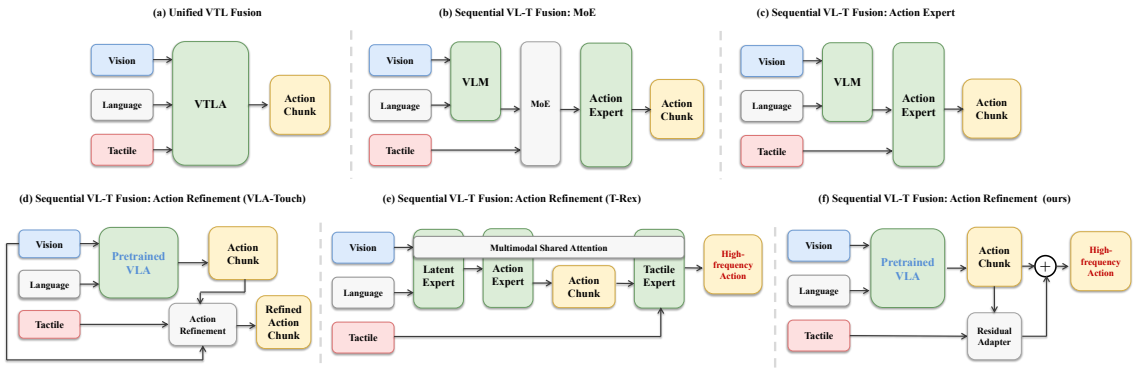

Fig. 2: Representative multimodal fusion strategies in existing VTLA models. To our knowledge, only VT-Bridge (ours) retains pretrained VLA backbones without architectural modification while enabling high-frequency tactile adaptation.

C. Slow–Fast Inference Scheme

Visual and force sensing are distinct in sensing characteristics yet functionally complementary [39]. Visual observations are high-dimensional, captured at a relatively low frequency, and informative of global scene context, whereas force observations are low-dimensional, available at a high frequency, and tightly coupled to local contact dynamics. To accommodate these modality-specific characteristics, a slow–fast inference scheme was introduced in RDP [40], in which a slow policy predicts latent action chunks from vision, while a fast decoder uses the latest tactile feedback to generate executable actions. This scheme effectively leverages the respective strengths of both modalities. Nevertheless, slow–fast inference has seen limited adoption in VTLA models, with only a few studies, including T-Rex [16] (Fig. 2(e)), FAVLA [17], and UniTacVLA [25].

III. Methods

To address the aforementioned limitations, this work proposes VT-Bridge, a lightweight residual adapter that augments pretrained VLA models with tactile-conditioned action adaptation for contact-rich manipulation. VT-Bridge retains the original pretrained VLA policy without modifying its structure and introduces a residual adapter working at a higher rate on top of its predicted action chunks, as illustrated in Fig. 1.

A. Low-frequency Pretrained VLA Backbone

The pretrained VLA backbone provides general-purpose visual-language understanding and guides the robot to complete the target manipulation task [2], [3]. Given the current observation $\boldsymbol{o}$ and language instruction $\boldsymbol{l}$, the VLA policy π^{VLA} predicts an action chunk:

$$\boldsymbol{A} \sim \pi^{\mathrm{VLA}}(\cdot \mid \boldsymbol{o}, \boldsymbol{l}), \tag{1}$$

where $\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_H]$ contains H future actions. This predicted chunk captures the coarse, task-level motion and provides the nominal action sequence subsequently refined by the residual adapter.

B. High-Frequency Residual Adapter

The residual adapter refines the nominal motion predicted by the VLA backbone using tactile feedback. As depicted in Fig. 1, we adopt a lightweight architectural design to facilitate faster inference and online adaptation to varying contact conditions.

Specifically, the residual adapter consists of two parallel transformer encoders that separately process the VLA-predicted action chunk and latest force observations: (i) The action encoder processes the embedded action chunk and produces an action feature $\boldsymbol{z}^a$ that represents the VLA's coarse motion intention and the overall action trend within the chunk. This feature is computed once and reused throughout chunk execution. (ii) In parallel, the force encoder extracts temporal contact features $\boldsymbol{z}_t^f$ from the latest force window:

$$\boldsymbol{F}_t = [\boldsymbol{f}_{t-n+1}, \boldsymbol{f}_{t-n+2}, \ldots, \boldsymbol{f}_t], \tag{2}$$

where n denotes the window size, and t indicates the current time step. The encoded action feature $\boldsymbol{z}^a$ and force feature $\boldsymbol{z}_t^f$ are then concatenated and passed to a multilayer perceptron decoder to predict the residual action:

$$\boldsymbol{a}_t^{\mathrm{res}} = \mathrm{MLP}\left([\boldsymbol{z}^a; \boldsymbol{z}_t^f]\right). \tag{3}$$

C. Inference and Execution

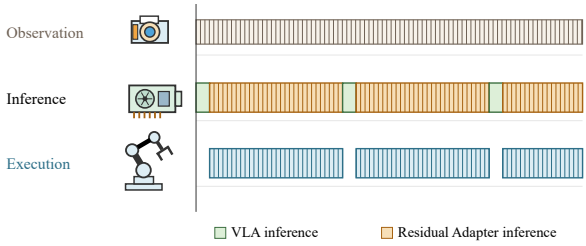

Fig. 3: Temporal scheduling of the VLA backbone, residual adapter, and robot executor.

During deployment, the VLA backbone and the residual adapter operate on the same GPU in a temporally interleaved manner, with each action chunk serving as the basic scheduling unit. As illustrated in Fig. 3, at the beginning of each cycle, the VLA backbone predicts an action chunk $\boldsymbol{A}$ via

Eq. (1). The backbone then becomes idle, and VT-Bridge enters the adaptation-and-execution loop. At each execution step t, the residual adapter predicts an action adaptation from the complete action chunk and the latest force window:

$$\boldsymbol{a}_t^{\mathrm{res}} = \pi^{\mathrm{res}}(\cdot \mid \boldsymbol{A}, \boldsymbol{F}_t). \tag{4}$$

The predicted residual action $\boldsymbol{a}_t^{\mathrm{res}}$ is then added to the corresponding nominal VLA action $\boldsymbol{a}_t$, yielding the adapted action $\boldsymbol{a}_t^{\mathrm{exe}}$ that is sent to the robot for execution.

$$\boldsymbol{a}_t^{\mathrm{exe}} = \boldsymbol{a}_t + \boldsymbol{a}_t^{\mathrm{res}}. \tag{5}$$

This loop continues until all actions in the predicted chunk have been executed. The residual adapter then pauses, and the VLA backbone is invoked to predict the next action chunk. In this manner, the two inference processes do not overlap and therefore do not contend for GPU computation.

IV. Training

The training pipeline of our method consists of four steps.

A. Teleoperated Demonstration Collection

Initially, we collect demonstrations for contact-rich manipulation tasks using the bilateral teleoperation strategy introduced in SharedAssembly [37]. By reflecting follower-side interaction loads to the leader robot, this strategy enables the operator to perceive contact changes and regulate the applied force throughout the demonstration. During each demonstration, we synchronously record the camera stream, the measured robot kinematic state and its corresponding desired command, and tactile signals in the form of the estimated external end-effector force and external joint torques.

B. VLA Backbone Fine-Tuning

Afterwards, we adapt the pretrained VLA backbone to the target manipulation tasks using the collected demonstrations. The camera observations, language instructions and robot states are converted into the training format required by each backbone, following its original observation preprocessing and action representation. We then fine-tune the backbone with LoRA [41]. The resulting task-adapted backbone produces the nominal action chunks used for residual dataset construction in the next step.

C. Residual Dataset Construction

Then, we run the fine-tuned VLA backbone on the recorded observations to generate nominal action chunks. After temporal alignment, the target residual action at each execution step k is computed as:

$$\boldsymbol{a}_k^{\mathrm{res}} = \boldsymbol{a}_k^{\mathrm{desired}} - \boldsymbol{a}_k, \tag{6}$$

where $\boldsymbol{a}_k^{\mathrm{desired}}$ denotes the desired command of the teleoperated robot's controller, and $\boldsymbol{a}_k$ is the corresponding action predicted by the fine-tuned VLA backbone. Using the desired command rather than the executed action preserves the operator's intended motion under contact. The effect of this choice is evaluated in an ablation study in Sec. VI-C.

Each residual target is paired with the corresponding nominal action chunk $\boldsymbol{A}$ and contact-feedback window $\boldsymbol{F}_k$, which serve as the inputs to the residual adapter. Following π_0 [2], VT-Bridge supports both Cartesian- and joint-space representations. To ensure consistency between the action and tactile representations, Cartesian actions are paired with estimated external end-effector force, whereas joint actions use external joint torque as tactile feedback.

Besides, we set the target residual action to zero for all non-contact samples. This encourages the adapter to preserve the VLA backbone's nominal decisions during free-space motion and focus its adaptation on regulating physical interactions after contact is established.

D. Residual Adapter Training

Finally, we freeze the fine-tuned VLA backbone and optimize only the residual adapter on the constructed residual dataset. This decoupled training preserves the nominal task behavior of the VLA backbone while enabling contact-conditioned residual adaptation.

V. Experiments

In this section, we conduct a comprehensive suite of real-world contact-rich manipulation experiments to address the following research questions:

- **Q1:** Can a lightweight residual adapter improve pre-trained foundation VLAs' performance in real-world contact-rich manipulation tasks?
- **Q2:** How does VT-Bridge overcome the last-inch bottleneck between geometrical near-success and functional task completion?
- **Q3:** How does using desired poses rather than executed poses to construct residual targets affect manipulation performance?

A. Experimental Setup

TABLE I: Configuration Details

Setting	Value
Demonstration sampling rate	30 Hz
Execution frequency	30 Hz
Joint stiffness matrix	$\mathrm{diag}(300, 300, 300, 200, 100, 60, 30)$
Force-history window n	50
Action-chunk horizon H	50
VLA fine-tuning steps	$30,000$
VLA fine-tuning batch size	32
VLA fine-tuning time	~ 8 h
π_0 chunk inference	153.1 ms
$\pi_{0.5}$ chunk inference	163.8 ms
SmolVLA chunk inference	155.4 ms
Action-chunk encoding	0.4 ms
Residual action prediction	0.7 ms

1) Robot Platform: The experimental platform comprises a Franka Emika Panda manipulator and two Intel RealSense D435i cameras. The robot controller runs on a NUC with an Intel i7-10700 CPU, while the VLA model and the proposed residual policy are executed on a separate workstation equipped with an NVIDIA RTX 3090 GPU. The foundation VLA models are fine-tuned using four NVIDIA A100-SXM4 GPUs. The concrete configurations in training and inference are summarized in Table I.

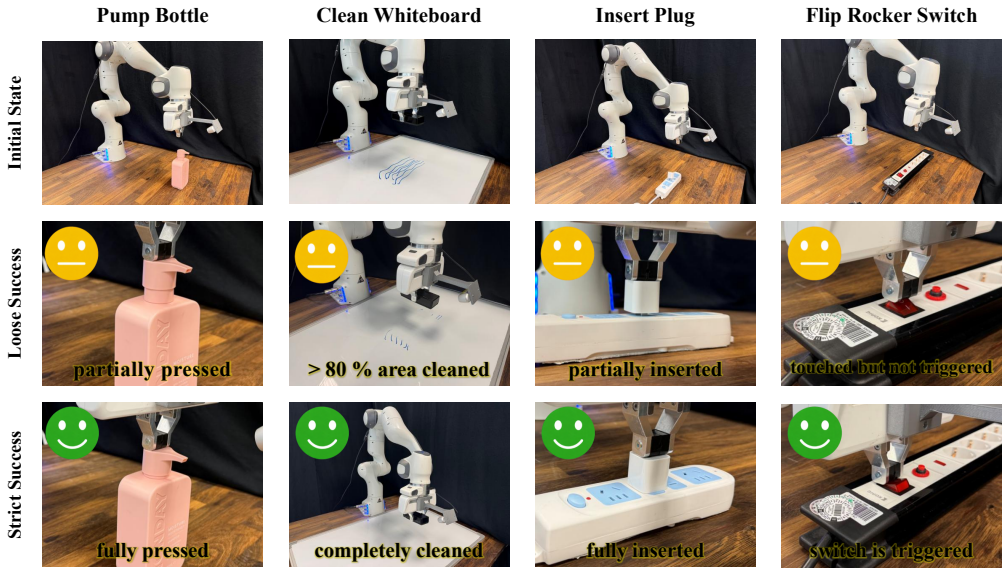

Fig. 4: Overview of contact-rich manipulation tasks, showing representative initial states and examples satisfying the loose and strict success criteria.

2) Evaluation Tasks: As illustrated in Fig. 4, the proposed method is evaluated on four contact-rich manipulation tasks: Pump Bottle, Clean Whiteboard, Insert Plug, and Flip Rocker Switch. These tasks cover diverse forms of physical interaction, including pressing, sustained surface contact, high-precision insertion, and discrete force-triggered actuation.

B. Evaluation Metrics

Each task is evaluated using two success criteria. The **strict success criterion** corresponds to complete task execution, while the **loose success criterion** indicates that the policy reaches the task-relevant interaction state and establishes the intended contact, but does not necessarily complete the task. This distinction separates failures in reaching the intended contact from failures in the subsequent contact-rich manipulation stage, enabling a more fine-grained evaluation of policy performance. Table II summarizes the corresponding definitions.

TABLE II: Success Criteria

Task	Strict Criterion	Loose Criterion
Pump Bottle	Full press	Press only
Clean Whiteboard	Complete cleaning	$>80\%$ area cleaned
Insert Plug	Full insertion	$>80\%$ insertion depth
Flip Rocker Switch	Switch triggered	Contact only

For each task, we report the strict success rate $\mathrm{SR}_{\mathrm{strict}}$, the loose success rate $\mathrm{SR}_{\mathrm{loose}}$, and the conditional completion rate ρ_{sl}:

$$\begin{aligned} \mathrm{SR}_{\mathrm{strict}} &= N_{\mathrm{strict}}/N\ (\%), \\ \mathrm{SR}_{\mathrm{loose}} &= N_{\mathrm{loose}}/N\ (\%), \\ \rho_{\mathrm{sl}} &= N_{\mathrm{strict}}/N_{\mathrm{loose}}\ (\%), \end{aligned} \tag{7}$$

where N denotes the total number of trials, while N_{strict} and N_{loose} denote the numbers of trials satisfying the strict and loose success criteria, respectively. ρ_{sl} measures the probability of achieving complete task execution once the policy has already reached a near-success state. It therefore provides a focused measure of the policy's ability to complete the contact-intensive stage of the task.

C. Compared Methods

We evaluate our approach on three representative foundation VLA backbones, namely π_0 [2], $\pi_{0.5}$ [3], and SmolVLA [38]. For each backbone, we compare three policy variants to isolate the effect of tactile integration.

1) **Fine-tuned**: the original VLA model fine-tuned with LoRA [41].
2) **VLA-Touch** [26]: we adopt the *VLA Action Refinement with Touch* module proposed in VLA-Touch to augment the fine-tuned VLA backbone.
3) **VT-Bridge** (ours): we employ the proposed lightweight residual adapter to augment the fine-tuned VLA backbone with the ability to regulate contact dynamics.

For fair comparison, all methods use the same set of demonstrations and identical training settings unless otherwise specified.

D. Experimental Procedure

For each manipulation task, we collect 50 human demonstrations via bilateral teleoperation [37] to construct the target manipulation dataset. For each pretrained VLA backbone, the model is first fine-tuned independently using LoRA on the collected demonstrations, serving as the Fine-tuned baseline. Based on the corresponding fine-tuned backbone, VLA-Touch [26] and our VT-Bridge are trained independently using the same demonstrations. The resulting backbone–method combinations are then evaluated on all four manipulation tasks. For each task, every combination is evaluated in 20 independent trials. Each trial is terminated when the strict success criterion is satisfied or the 90 s time limit is reached.

TABLE III: Strict Success Rate (%)

Backbone	Method	Pump Bottle	Clean Whiteboard	Insert Plug	Flip Rocker Switch	Average
π_0 [2]	Fine-tuned [41]	0.0 (0/20)	5.0 (1/20)	0.0 (0/20)	20.0 (4/20)	6.3
	VLA-Touch [26]	0.0 (0/20)	5.0 (1/20)	0.0 (0/20)	50.0 (10/20)	13.8
	VT-Bridge (ours)	**85.0 (17/20)**	**20.0 (4/20)**	**70.0 (14/20)**	**85.0 (17/20)**	**65.0**
$\pi_{0.5}$ [3]	Fine-tuned [41]	0.0 (0/20)	35.0 (7/20)	0.0 (0/20)	55.0 (11/20)	22.5
	VLA-Touch [26]	0.0 (0/20)	20.0 (4/20)	5.0 (1/20)	55.0 (11/20)	20.0
	VT-Bridge (ours)	**90.0 (18/20)**	**40.0 (8/20)**	**70.0 (14/20)**	**75.0 (15/20)**	**68.8**
SmolVLA [38]	Fine-tuned [41]	0.0 (0/20)	10.0 (2/20)	0.0 (0/20)	15.0 (3/20)	6.3
	VLA-Touch [26]	0.0 (0/20)	5.0 (1/20)	0.0 (0/20)	25.0 (5/20)	7.5
	VT-Bridge (ours)	**70.0 (14/20)**	**45.0 (9/20)**	**65.0 (13/20)**	**40.0 (8/20)**	**55.0**

TABLE IV: Conditional Completion Rate ρ_{sl} (%)

Backbone	Method	Pump Bottle	Clean Whiteboard	Insert Plug	Flip Rocker Switch	Average
π_0 [2]	Fine-tuned [41]	0.0 (0/19)	5.3 (1/19)	0.0 (0/13)	23.5 (4/17)	7.2
	VLA-Touch [26]	0.0 (0/5)	5.9 (1/17)	0.0 (0/16)	66.7 (10/15)	18.1
	VT-Bridge (ours)	**89.5 (17/19)**	**20.0 (4/20)**	**93.3 (14/15)**	**100.0 (17/17)**	**75.7**
$\pi_{0.5}$ [3]	Fine-tuned [41]	0.0 (0/20)	35.0 (7/20)	0.0 (0/16)	**100.0 (11/11)**	33.8
	VLA-Touch [26]	0.0 (0/10)	20.0 (4/20)	6.3 (1/16)	84.6 (11/13)	27.7
	VT-Bridge (ours)	**94.7 (18/19)**	**40.0 (8/20)**	**77.8 (14/18)**	**100.0 (15/15)**	**78.1**
SmolVLA [38]	Fine-tuned [41]	0.0 (0/19)	10.5 (2/19)	0.0 (0/8)	16.7 (3/18)	6.8
	VLA-Touch [26]	0.0 (0/4)	6.3 (1/16)	0.0 (0/10)	29.4 (5/17)	8.9
	VT-Bridge (ours)	**87.5 (14/16)**	**45.0 (9/20)**	**86.7 (13/15)**	**61.5 (8/13)**	**70.2**

VI. Experimental Results

A. Overall Performance on Contact-Rich Manipulation (Q1)

As illustrated in Fig. 5, VT-Bridge substantially outperforms baseline methods across all three representative VLA backbones. Compared with directly fine-tuning the backbones on the same robot demonstrations, VT-Bridge increases the overall average strict success rate from 11.7% to 62.9%, yielding an absolute improvement of 51.2 percentage points. The consistently substantial gains across pretrained VLA models demonstrate the broad effectiveness and applicability of VT-Bridge across various pretrained VLA backbones.

To further examine the performance across different contact-rich manipulation scenarios, Table III reports the strict success rate for each individual task. Notably, VT-Bridge achieves the highest strict success rate in all task–backbone combinations, without exception. This remarkably consistent advantage highlights the robustness and broad applicability of the proposed residual adaptation framework.

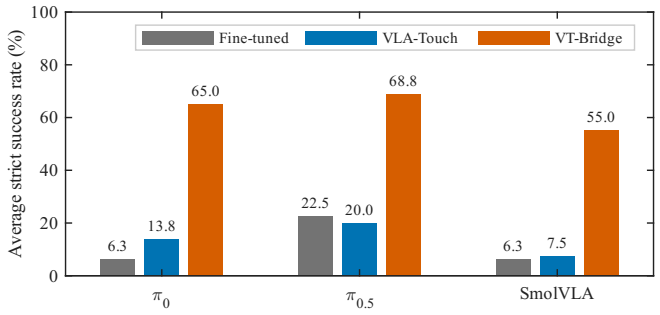

Fig. 5: Average strict success rates across VLA backbones and real-world contact-rich manipulation tasks.

B. Last-Inch Bottleneck in Contact-Rich Manipulation (Q2)

Our experiments reveal that a notable proportion of failure cases do not result from an incorrect task intention. Instead, the policies often approach task completion but fail at the final contact-sensitive step. For example, the robot may successfully depress the pump head but not far enough to dispense liquid, or partially insert the plug without fully seating it. Although these executions are geometrically close to success, they do not achieve the intended functional outcome and must therefore be considered failures in real-world manipulation. We refer to this gap between geometrical near-success and functional completion as the **last-inch bottleneck in contact-rich manipulation**.

To quantify a policy's ability to overcome this bottleneck, we analyze the conditional completion rate ρ_{sl}, introduced in Sec. V-B. By conditioning on loose success, this metric reduces the influence of backbone-dependent differences in task understanding and motion generation, enabling a more focused evaluation of tactile adaptation during last-inch completion.

As depicted in Table IV, VT-Bridge achieves the highest conditional completion rate in all task–backbone combinations. On average, VT-Bridge increases the macro-averaged conditional completion rate by 58.8 percentage points over fine-tuned VLA models, demonstrating a substantially stronger ability to overcome the last-inch bottleneck. The Pump Bottle task provides a striking example: none of the baseline–backbone combinations successfully converted a geometrical near-success into functional completion, as they failed to fully depress the pump head. In contrast, VT-Bridge achieves an average conditional completion rate of 90.6%. A similar trend is observed in the Insert Plug task, where fine-tuned VLAs remain at 0% across all three backbones, and VLA-Touch reaches only 6.3% in one combination, while

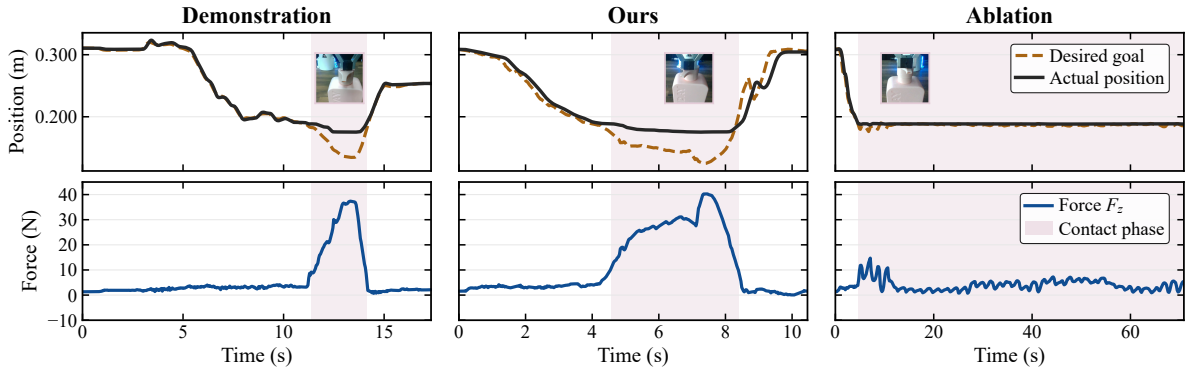

Fig. 6: Interaction profiles along the z-axis for the Pump Bottle task.

VT-Bridge achieves an average of 85.9%. These results show that our method specifically improves the final transition from geometric near-success to functional completion.

C. Ablation Study on Residual Target Construction: Desired vs. Executed Poses (Q3)

As introduced in Sec. IV-C, our default residual targets are computed as the difference between the desired poses $\boldsymbol{a}^{\text{desired}}$ recorded in the demonstrations and the nominal poses $\boldsymbol{a}$ predicted by the fine-tuned VLA. For this ablation, we construct an alternative residual dataset by replacing the desired poses with the corresponding executed poses $\boldsymbol{a}^{\text{executed}}$, while keeping all other settings unchanged. As shown in Fig. 7, residual targets constructed from desired poses consistently outperform those constructed from executed poses across all task–backbone combinations. On average, using desired poses increases the strict success rate from 12.1% to 62.9%.

To support safe physical interaction, we use impedance control throughout our experiments. In tasks requiring high interaction forces, controller compliance creates a pronounced offset between the desired and executed poses. As illustrated in Fig. 6, fully depressing the pump head requires a peak interaction force of approximately 40 N in both the demonstration and the successful VT-Bridge trial. Correspondingly, the desired–executed pose offset reaches approximately 4–5 cm. Residual targets constructed from desired poses explicitly incorporate this force-related offset into the training data, whereas those constructed from executed poses omit it. Consequently, even if the ablation policy accurately predicts the executed pose recorded in the demonstration, issuing that pose as a new desired command produces another tracking shortfall under contact. The robot guided by the ablation policy therefore fails to reproduce the demonstrated pose and contact force. Although employing a high-stiffness controller could reduce this offset, it would also increase the risk of excessive interaction forces [42], [43]. By contrast, our design improves task reliability while maintaining safe physical interaction.

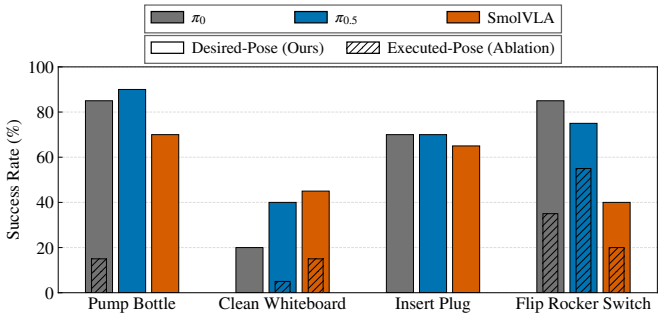

Fig. 7: Ablation of residual target construction. Strict success rates obtained using desired-pose (ours) and executed-pose (ablation) to construct the residual dataset.

A similar pattern is observed in Insert Plug, where the peak interaction force also approaches 40 N. By contrast, Flip Rocker Switch requires only around 7 N to accomplish the task, resulting in a smaller compliance-induced offset. The executed-pose ablation therefore performs better on this task than on the two higher-force tasks. Nevertheless, constructing residual targets from desired poses still yields an 82% relative improvement over the ablation policy on this task. Overall, using desired poses $\boldsymbol{a}^{\text{desired}}$ to construct residual targets consistently improves performance across evaluated contact-rich tasks, with more pronounced gains as the required interaction force increases.

VII. Conclusion

This work presents VT-Bridge, a resource-efficient residual adaptation strategy that bridges pretrained foundation VLAs to tactile-aware VTLAs for contact-rich manipulation. Rather than training a VTLA model from scratch or modifying the pretrained VLA backbone architectures, VT-Bridge employs an identical lightweight residual-adapter architecture across pretrained VLAs and uses backbone-specific weights to refine actions at the robot execution frequency. This design retains the vision-language understanding and motion-generation capabilities acquired through large-scale VLA pretraining while enabling responsive contact regulation. Using 50 vision-tactile demonstrations per task, VT-Bridge consistently improved three representative VLA backbones across four real-world contact-rich manipulation tasks, raising the average task completion rate from 11.7% to 62.9%. These experimental results demonstrate its effectiveness and broad applicability, showcasing a promising pathway for developing VTLAs from existing pretrained VLAs. Future work will focus on improving the representation and adaptation capabilities of

the residual adapter to achieve stronger and more consistent performance gains.

ACKNOWLEDGMENT

As non-native English speakers, the authors used GPT-5.6 for grammatical and linguistic refinement. Codex was also used to organize the codebase for open-source release. All AI-assisted suggestions and modifications were reviewed and verified by the authors.

REFERENCES

[1] R. Firoozi, J. Tucker, S. Tian *et al.*, "Foundation models in robotics: Applications, challenges, and the future," *The International Journal of Robotics Research*, vol. 44, no. 5, pp. 701–739, 2025.

[2] K. Black, N. Brown, D. Driess, A. Esmail *et al.*, "π_0: A vision-language-action flow model for general robot control," *arXiv preprint arXiv:2410.24164*, 2024.

[3] K. Black, N. Brown, J. Darpinian, K. Dhabalia *et al.*, "$\pi_{0.5}$: A vision-language-action model with open-world generalization," in *Proceedings of The 9th Conference on Robot Learning (CoRL)*. PMLR, 2025, pp. 17–40.

[4] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar *et al.*, "Rt-1: Robotics transformer for real-world control at scale," in *Robotics: Science and Systems (RSS)*, 2023.

[5] C. Chi, S. Feng, Y. Du *et al.*, "Diffusion policy: Visuomotor policy learning via action diffusion," in *Robotics: Science and Systems (RSS)*, 2023.

[6] D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch *et al.*, "Palm-e: An embodied multimodal language model," in *International Conference on Machine Learning (ICML)*. PMLR, 2023, pp. 8469–8488.

[7] Y. Guo, Y. Hu, J. Zhang, Y.-J. Wang *et al.*, "Prediction with action: Visual policy learning via joint denoising process," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[8] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao *et al.*, "Openvla: An open-source vision-language-action model," in *Conference on Robot Learning (CoRL)*, 2024.

[9] M. J. Kim, C. Finn, and P. Liang, "Fine-tuning vision-language-action models: Optimizing speed and success," *arXiv preprint arXiv:2502.19645*, 2025.

[10] J. Liu, H. Chen, P. An, Z. Liu *et al.*, "Hybridvla: Collaborative diffusion and autoregression in a unified vision-language-action model," *arXiv preprint arXiv:2503.10631*, 2025.

[11] J. Bjorck, F. Castañeda, N. Cherniadev *et al.*, "Gr00t n1: An open foundation model for generalist humanoid robots," *arXiv preprint arXiv:2503.14734*, 2025.

[12] D. Ghosh, H. Walke, K. Pertsch, K. Black *et al.*, "Octo: An open-source generalist robot policy," in *Robotics: Science and Systems (RSS)*, 2024.

[13] K. Pertsch, K. Stachowicz, B. Ichter, D. Driess *et al.*, "FAST: Efficient action tokenization for vision-language-action models," in *Robotics: Science and Systems (RSS)*, 2025.

[14] Z. Zhou, Y. Zhu, M. Zhu, J. Wen *et al.*, "Chatvla: Unified multimodal understanding and robot control with vision-language-action model," in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2025, pp. 5377–5395.

[15] B. Zitkovich, T. Yu, S. Xu *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," in *Conference on Robot Learning (CoRL)*. PMLR, 2023, pp. 2165–2183.

[16] D. Niu, Z. Liu, Z. Wang, B. Shao, Z.-H. Yin *et al.*, "T-rex: Tactile-reactive dexterous manipulation," *arXiv preprint arXiv:2606.17055*, 2026.

[17] Y. Li, P. Tang, W. Zhang, C. Zhu, Y. Duan *et al.*, "Favla: A force-adaptive fast-slow vla model for contact-rich robotic manipulation," *arXiv preprint arXiv:2602.23648*, 2026.

[18] Y. Zhang, Y. Wang, X. Sun, K. Huang *et al.*, "Craft: Adapting vla models to contact-rich manipulation via force-aware curriculum fine-tuning," *arXiv preprint arXiv:2602.12532*, 2026.

[19] R. Zhao, W. Wang, Y. Ma, X. Li *et al.*, "Fd-vla: Force-distilled vision-language-action model for contact-rich manipulation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

[20] P. Zhang, B. Xie, X. Meng *et al.*, "Feeling the unexpected: Restacvla for contact-rich manipulation via residual tactile representation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2026.

[21] X. Li, M. Cai, J. Xu, J. Zhu *et al.*, "At-vla: Adaptive tactile injection for enhanced feedback reaction in vision-language-action models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

[22] Y. Li, Zhaxizhuoma, H. Jiang, J. Xia *et al.*, "Forcevla2: Unleashing hybrid force-position control with force awareness for contact-rich manipulation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

[23] J. Yu, H. Liu, Q. Yu, J. Ren, C. Hao *et al.*, "Forcevla: Enhancing vla models with a force-aware moe for contact-rich manipulation," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[24] Y. Wang, W. Chen, Z. Wen *et al.*, "Never too late for force: Accelerating vla post-training with reactive force injection," *arXiv preprint arXiv:2607.14236*, 2026.

[25] X. Zhang, Y. Zhang, J. Shi, F. Zhu *et al.*, "Unitacvla: Unified tactile understanding and prediction in vision language action models," *arXiv preprint arXiv:2606.31723*, 2026.

[26] J. Bi, K. Y. Ma, C. Hao, M. Z. Shou, and H. Soh, "Vla-touch: Enhancing vision-language-action model with dual-level tactile feedback," *IEEE Robotics and Automation Letters*, vol. 11, no. 7, pp. 8487–8494, 2026.

[27] C. Zhang, P. Hao, X. Cao, X. Hao, S. Cui, and S. Wang, "Vtla: Vision-tactile-language-action model with preference learning for insertion manipulation," *arXiv preprint arXiv:2505.09577*, 2025.

[28] H. Zhang, W.-H. Huang, Q. Tong, G. Solak *et al.*, "Compliantvla-adaptor: Vlm-guided variable impedance action for safe contact-rich manipulation," *arXiv preprint arXiv:2601.15541*, 2026.

[29] K. Gubernatorov, M. Sannikov, I. Mikhalchuk *et al.*, "Hapticvla: Contact-rich manipulation via vision-language-action model without inference-time tactile sensing," *arXiv preprint arXiv:2603.15257*, 2026.

[30] Z. Cheng, Y. Zhang, W. Zhang *et al.*, "Omnivtla: Vision-tactile-language-action model with semantic-aligned tactile sensing," *arXiv preprint arXiv:2508.08706*, 2025.

[31] Z. Zhang, H. Xu, Z. Yang *et al.*, "Elucidating the design space of torque-aware vision-language-action models," in *Proceedings of the 9th Conference on Robot Learning (CoRL)*, 2025.

[32] Z. Wang, Y. Wang, M. Ren *et al.*, "Tacmamba: A tactile history compression adapter bridging fast reflexes and slow vla reasoning," *arXiv preprint arXiv:2603.01700*, 2026.

[33] K. Zhang, H. Zhang, Z. Xu *et al.*, "Tacvla: Contact-aware tactile fusion for robust vision-language-action manipulation," *arXiv preprint arXiv:2603.12665*, 2026.

[34] J. Huang, S. Wang, F. Lin, Y. Hu, C. Wen, and Y. Gao, "Tactile-vla: Unlocking vision-language-action model's physical knowledge for tactile generalization," *arXiv preprint arXiv:2507.09160*, 2025.

[35] H.-S. Fang, H. Fang *et al.*, "Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 653–660.

[36] D. Sliwowski, S. Jadav, S. Stanovcic *et al.*, "Reassemble: A multimodal dataset for contact-rich robotic assembly and disassembly," *arXiv preprint arXiv:2502.05086*, 2025.

[37] Y. Wu, X. Chen, Y. Chen, H. Sadeghian, F. Wu, Z. Bing, and A. Knoll, "Sharedassembly: A data collection approach via shared tele-assembly," in *2026 IEEE Conference on Telepresence*. IEEE, 2026.

[38] M. Shukor, D. Aubakirova, F. Capuano, P. Kooijmans *et al.*, "Smolvla: A vision-language-action model for affordable and efficient robotics," *arXiv preprint arXiv:2506.01844*, 2025.

[39] B. Siciliano, O. Khatib, and T. Kröger, *Springer handbook of robotics*. Springer, 2008, vol. 200.

[40] H. Xue, J. Ren, W. Chen, G. Zhang, Y. Fang, G. Gu, H. Xu, and C. Lu, "Reactive diffusion policy: Slow-fast visual-tactile policy learning for contact-rich manipulation," *arXiv preprint arXiv:2503.02881*, 2025.

[41] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu *et al.*, "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022.

[42] N. Hogan, "Impedance control: An approach to manipulation: Part II—Implementation," 1985.

[43] N. Hogan and S. P. Buerger, "Impedance and interaction control," *Robotics and automation handbook*, vol. 1, 2005.