ForceVLA: Enhancing VLA Models with a Force-aware MoE for Contact-rich Manipulation

Jiawen Yu[1*], Hairuo Liu[2,3*], Qiaojun Yu[4,2†], Jieji Ren[2], Ce Hao[5], Haitong Ding[6], Guangyu Huang[7], Guofan Huang[1], Yan Song[1], Panpan Cai[2,3], Wenqiang Zhang[1†], Cewu Lu[2,3,8†]
[1] Fudan University, [2]Shanghai Jiao Tong University, [3]Shanghai Innovation Institute, [4]Shanghai AI Lab, [5] National University of Singapore, [6] Shanghai University, [7] Xi'an Jiaotong University, [8] Noematrix Intelligence
[*] Equal contribution [†] Corresponding authors

Abstract

Vision-Language-Action (VLA) models have advanced general-purpose robotic manipulation by leveraging pretrained visual and linguistic representations. However, they struggle with contact-rich tasks that require fine-grained control involving force, especially under visual occlusion or dynamic uncertainty. To address these limitations, we propose **ForceVLA**, a novel end-to-end manipulation framework that treats external force sensing as a first-class modality within VLA systems. ForceVLA introduces **FVLMoE**, a force-aware Mixture-of-Experts fusion module that dynamically integrates pretrained visual-language embeddings with real-time 6-axis force feedback during action decoding. This enables context-aware routing across modality-specific experts, enhancing the robot's ability to adapt to subtle contact dynamics. We also introduce **ForceVLA-Data**, a new dataset comprising synchronized vision, proprioception, and force-torque signals across five contact-rich manipulation tasks. ForceVLA improves average task success by 23.2% over strong π_0-based baselines, achieving up to 80% success in tasks such as plug insertion. Our approach highlights the importance of multimodal integration for dexterous manipulation and sets a new benchmark for physically intelligent robotic control. Code and data will be released at website.

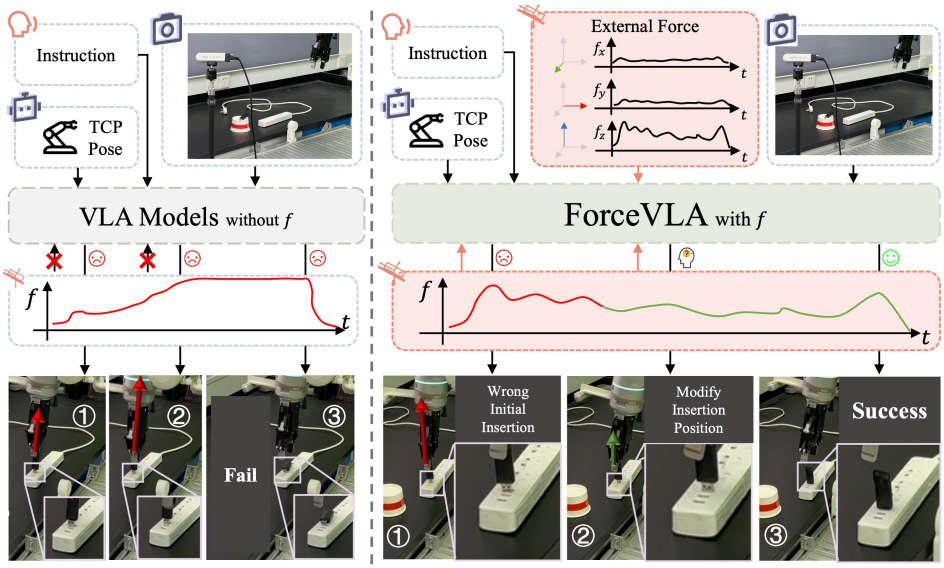
1 Introduction

Robotic learning has advanced rapidly with the rise of embodied AI, driven by large-scale manipulation datasets and the emergence of foundation models [1, 2, 3]. These models exhibit strong adaptability, enabling rapid generalization to novel tasks with minimal supervision [4, 5, 6]. In parallel, Vision-Language Models (VLMs) have achieved remarkable generalization through large-scale multimodal alignment [7, 8], learning transferable representations that support a wide range of downstream tasks.

Building on this progress, OpenVLA [9] introduced Vision-Language-Action (VLA) models to bridge perception and control for real-world robotic manipulation. By leveraging VLM-based encoders, these models demonstrate strong performance in semantic grounding, language following, and zero-shot generalization. π_0 [10] further enhances this framework using stronger VLM backbones [11] and flow-based action generation, showing that pretrained multimodal VLA models can acquire robust physical-world priors and can be fine-tuned efficiently with only a few demonstrations.

However, contact-rich manipulation demands more than semantic grounding and spatial planning—it is fundamentally driven by interaction forces [12, 13]. Existing VLA models rely heavily on visual and linguistic cues, often overlooking force sensing, a modality critical for precise physical interaction. In contrast, humans naturally integrate tactile and proprioceptive feedback to adapt their manipulation

39th Conference on Neural Information Processing Systems (NeurIPS 2025).



(a) VLA Models w/o force input (b) Our ForceVLA with force modality input

Figure 1: Comparison between ForceVLA and baselines without force input. Without force feedback, the policy fails to correct pose errors and completes insertion incorrectly. In contrast, ForceVLA leverages external force signals to adjust insertion strategies dynamically, leading to successful execution despite initial misalignment.

strategies [14]. As a result, VLA models frequently struggle with tasks such as insertion, tool use, or assembly—especially under occlusion or poor visual conditions—leading to brittle behavior or task failure. Moreover, force requirements evolve across different task phases: delicate grasping, controlled insertion, and compliant surface contact—each requiring distinct forms of force modulation. Current methods lack mechanisms to perceive and adapt to these dynamic variations, limiting their ability to reason over time about physical interactions.

To address these limitations, we introduce **ForceVLA**, a novel framework that augments VLA models with a force-aware Mixture-of-Experts (MoE) module, enabling effective reasoning and context-sensitive, force-informed action generation in contact-rich manipulation tasks, as illustrated in Figure 1. ForceVLA is grounded in the key insight that 6D external force sensed at the robot's end-effector, should be treated as a first-class modality and formally integrated into the action expert module to enable phase-aware action generation based on force feedback during physical interaction. To realize this integration, ForceVLA incorporates a force-aware MoE module, named **FVLMoE**, designed to perform modality- and phase-aware fusion of visual-linguistic representations with real-time force feedback from embodied interaction during action planning. Through a gating mechanism, FVLMoE computes dynamic routing weights over expert subnetworks, each specialized for different modalities across task execution phases. By adaptively activating these experts based on high-level task instructions and low-level interaction feedback, ForceVLA captures subtle yet critical, phase-dependent variations during physical interaction and generates precise, phase-aligned, and force-aware action chunking. To facilitate research in force-aware manipulation, we introduce **ForceVLA-Data**, a carefully curated dataset comprising synchronized vision, proprioception, and 6-axis force-torque signals collected across five representative contact-rich manipulation tasks. This dataset addresses the scarcity of publicly available force-annotated robotic manipulation data and provides a valuable resource for training and evaluating multi-modal robotic learning systems. Our main contributions are:

- We present a novel framework that integrates force, vision, language, and action for improved precision and stability on contact-rich manipulation tasks. Key to our approach is a force-aware Mixture-of-Experts-based fusion module, which enables dynamic processing and deep integration of force, visual, and language features during action generation, significantly enhancing physical interaction capabilities in VLA systems.
- We introduce a new dataset featuring synchronized vision, proprioception, and force-torque signals across five contact-rich manipulation tasks. We also provide a complete data collection

pipeline—including teleoperation tools and data converters—and commit to open-sourcing all resources to advance research in force-aware robotic manipulation.

- Through experiments on five challenging tasks, ForceVLA achieves up to 80% task success and improves average performance by 23.2% over baselines, demonstrating strong generalization to novel objects, occlusions, and physical perturbations.

2 Related Works

Robotic VLA domain. Recent research in Vision-Language-Action (VLA) models has focused on leveraging large-scale multimodal pretraining to generalize robotic policies across tasks and embodiments [4, 6, 9, 15, 16, 17, 18, 19, 20]. These models typically map visual and language inputs to low-level control signals via end-to-end learning. Flow-based architectures such as π_0 [10, 21] integrate pretrained vision-language encoders with fast action decoders to achieve high-frequency outputs. Other works incorporate reasoning mechanisms [22, 23, 24, 25], action space compression, or 3D point cloud inputs [26] to improve instruction grounding and task execution. Diffusion-based models [5, 27, 28, 29, 30] introduce stochastic generation for diverse, long-horizon behaviors, though they often incur high training and inference costs. Despite these advances, most VLA approaches remain limited to vision and language inputs, making them less effective in contact-rich or occluded manipulation scenarios where tactile feedback is critical.

Contact-rich manipulation domain. Traditional vision-only methods struggle with dynamic interactions requiring fine-grained feedback. To address this, recent works integrate force sensing [31, 32, 33, 34], enabling improved motion stability and accuracy. Xie et al. [35, 13] provide foundational studies on the role of force feedback in robotic control. Tactile sensing has also emerged as a powerful modality: TLA [36] and Tac-Man [37] demonstrate enhanced performance in fine manipulation and articulation tasks. VTLA [38] explores sim-to-real transfer using fingertip tactile sensors with a new simulation dataset. In contrast to VTLA's simulation-first approach, our work adopts a real-world-first paradigm with 6D force-torque sensing and introduces a novel MoE architecture for online force modulation rather than focusing on sim-to-real transfer. Multimodal fusion methods [39, 40] show promise in complex environments. FuSe [41] proposes a finetuning recipe to adapt existing generalist models to incorporate new modalities. While FuSe focuses on adaptation strategies, our contribution lies in designing a complete end-to-end policy architecture with deep integration of force feedback through an MoE-based routing mechanism. However, current approaches are often limited to static modality fusion and lack dynamic routing or unified modeling frameworks. Furthermore, few evaluate cross-task generalization in real-world contact-rich settings.

Contact-rich manipulation datasets. Large-scale datasets have been instrumental in advancing robotic learning. RH20T [42] provides a valuable collection of diverse manipulation demonstrations, offering raw data with basic reading tools. Factory [43] introduces a powerful simulation toolkit for contact-rich interactions at scale, providing virtual assets and controllers for policy training in simulated environments. IndustReal [44] demonstrates an approach that explicitly avoids force-torque sensors, instead relying on alternative cues to manage contact during manipulation. While these pioneering efforts have advanced the field in task diversity, simulation capabilities, and sensor-free approaches, they have limitations that our work addresses. Specifically, existing datasets often lack out-of-the-box usability for modern training frameworks, rely on simulation that introduces a sim-to-real gap, or deliberately exclude explicit force sensing. In contrast, our ForceVLA-Data provides three unique contributions: (1) pre-processed data in the SOTA Lerobot format with conversion tools for plug-and-play compatibility, significantly reducing data-wrangling overhead; (2) a reality-first methodology with data collected entirely in the real world, bypassing the sim-to-real gap; and (3) deep integration of high-resolution 6D force-torque sensor data, positioning our dataset as an essential resource for developing and benchmarking policies that explicitly leverage force feedback for dexterous manipulation.

MoE architecture-related work. Mixture-of-Experts (MoE) architectures improve model scalability and efficiency by activating sparse expert subnetworks [45, 46, 47, 48, 49]. Follow-up work [50] improves MoE training stability and task transferability. In the multimodal domain, LIMOE [51] integrates sparse expert layers for joint vision-language learning. Recent applications in robotics [52, 29] adopt MoE layers within VLA models to enhance policy generalization and adaptability. However, these methods largely omit explicit modeling of the force/tactile modalities, and lack mechanisms for dynamically routing across multimodal signals in contact-intensive tasks.

3 Preliminary

Problem Formulation. Figure 2 shows the setting of robot manipulation tasks. The robot's observation at timestep t consists of base and hand visual inputs V_t^b and V_t^h, the proprioceptive state $s_t \in \mathbb{R}^7$, and a single-timestep external force-torque reading $f_t \in \mathbb{R}^6$, which are collectively denoted as $O_t = \{V_t^b, V_t^h, s_t, f_t\}$. Given a language instruction L, the objective is to learn an end-to-end policy $\pi(A_t|O_t, L)$ that outputs low-level, executable action chunk $A_t = \{a_t, a_{t+1}, ..., a_{t+H-1}\}$[10] maximizing the likelihood of completing the contact-rich task, where s_t is a vector of TCP pose concatenated with gripper width. TCP position is represented by Cartesian coordinates (x, y, z) and orientation is represented by Euler angles (α, β, γ). f_t is the estimated external wrench applied on TCP at timestep t and expressed in world frame, which consists of $\mathbb{R}^3$ force and $\mathbb{R}^3$ moment: $f_t = \{f_{tx}, f_{ty}, f_{tz}, m_{tx}, m_{ty}, m_{tz}\}$.

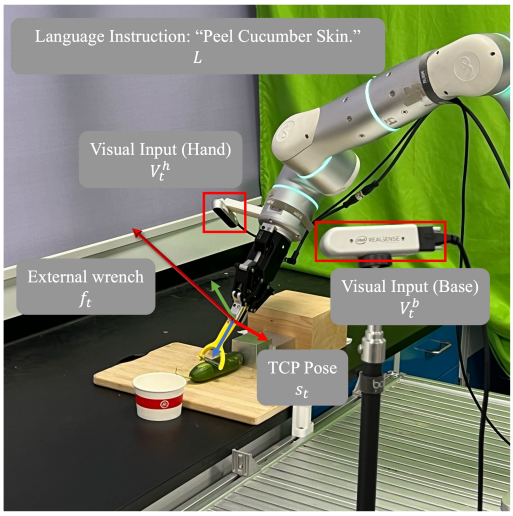


Figure 2: Robot manipulation tasks setting.

MoE Architecture. We select Mixture-of-Experts (MoE)[49, 47] as our fusion layer. The core idea is to distribute different modalities to a larger set of smaller, specialized "expert" subnetworks, only a fraction of which are activated for any given input token. An MoE layer typically comprises a set of N **expert networks**, denoted as $\{E_i\}_{i=1}^N$ and a **gating network** (also referred to as a router), denoted as G. This network takes an input token x and dynamically determines which of the N experts should process it. In prevalent sparse MoE implementations, for an input token x, the gating network $G(x)$ produces scores or logits that are used to select a small subset of k experts (typically $k = 1$ or $k = 2$, where $k \ll N$) from the total pool of N experts. The input token x is then routed only to these k active experts. The outputs of these active experts, $E_i(x)$, are subsequently aggregated, commonly through a weighted sum where the weights $g_i(x)$ are also derived from the gating network. The final output $y(x)$ of the MoE layer can be expressed as: $y(x) = \sum_{i \in \text{TopK}(G(x))} g_i(x) E_i(x)$, where $\text{TopK}(G(x))$ denotes the set of indices of the top-k experts selected by the gating network for input x.

4 ForceVLA

4.1 Overview of ForceVLA

ForceVLA is an end-to-end multimodal robotic policy designed for contact-rich manipulation. Its pipeline is illustrated in Figure 3. Building upon the π_0 framework [10], it integrates vision, language, proprioception, and 6-axis force feedback to generate actions through a conditional flow matching model [53, 54]. Visual inputs from multiple RGB cameras and task instructions are encoded by a SigLIP-based [55] vision-language model (based on PaliGemma [11]) into contextual embeddings. These embeddings, combined with proprioceptive and force cues, condition an iterative denoising process that predicts the action trajectory.

FVLMoE is the core module enabling effective force integration. Force readings are linearly projected into dedicated tokens and fused with vision-language embeddings via a Mixture-of-Experts (MoE) module. Inspired by MoE's strength in multi-task and modality-specific learning [56, 51], FVLMoE adaptively routes and processes multimodal inputs. Its output serves as a rich guidance signal for the flow model, allowing ForceVLA to handle subtle contact dynamics and visually ambiguous scenarios with greater precision and robustness.

4.2 FVLMoE Architecture

The FVLMoE module is specifically designed for the fusion of multimodal sensory information. Its design enables the model to integrate rich contextual understanding from vision and language with the immediate, fine-grained dynamics captured by force-torque sensors. This fusion is critical for enabling robust and adaptive behavior in contact-rich manipulation tasks. The architecture and operation of the FVLMoE can be detailed in the following stages:

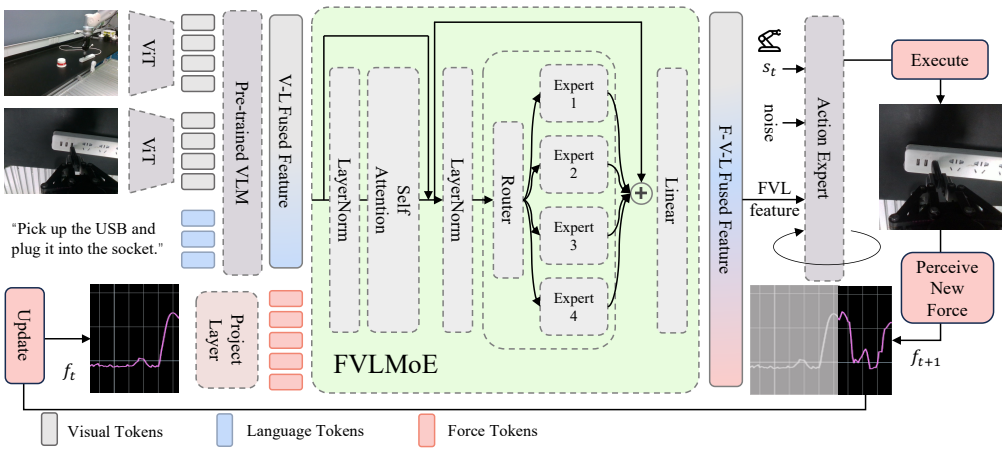


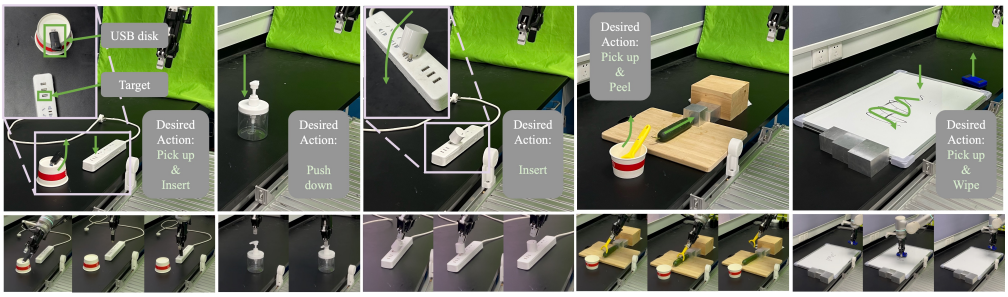
Figure 3: Overview of our ForceVLA model. Visual and language inputs are processed by a pre-trained VLM to form contextual embeddings. External force signals are projected and fused with VLM outputs via the FVLMoE module. The resulting multimodal features guide a flow-based action head to generate contact-aware robot actions.

Input Mapping for Multiple Modalities. How to determine the optimal stage and method for a novel force modality incorporation poses a significant design challenge. Following extensive experimentation, we established an approach where the force modality is introduced after the primary VLM has processed visual and linguistic inputs. Specifically, force features are fed as distinct inputs into the FVLMoE module, positioning it to function akin to a higher-level cortical association area responsible for integrating the VLM's pre-trained visual-linguistic representations with the newly introduced force tokens. This strategy contrasts with introducing force prior to, or concurrently with, the VLM's initial fusion of visual and language modalities. The empirical justification for this architectural decision is elaborated in the Ablation Studies section (Section 5.4).

The FVLMoE module, in line with this design choice, ingests a sequence of token embeddings E_{in} formed by the concatenation of visual-linguistic features and a dedicated force token. The VL features, denoted as $E_{VL} \in \mathbb{R}^{N_{VL} \times D_{model}}$, are outputs from the primary Vision-Language Model, encapsulating contextual understanding derived from processed image streams and textual instructions. Concurrently, the raw 6-axis force-torque sensor data, $f_{raw} \in \mathbb{R}^6$, is transformed by a linear projection ϕ_F into a force token embedding $E_F = \phi_F(f_{raw}) \in \mathbb{R}^{D_{model}}$. The final input to the FVLMoE is thus the concatenated sequence $E_{in} = [E_{VL}; E_F] \in \mathbb{R}^{(N_{VL}+1) \times D_{model}}$, where the force token is appended to the visual-linguistic context for subsequent joint processing within the MoE architecture.

Multimodal Routing and Fusion Computation. Once the combined multimodal sequence $E_{in} \in \mathbb{R}^{(N_{VL}+1) \times D_{model}}$ is formed, it undergoes hierarchical processing within the FVLMoE module. E_{in} is passed through an encoder layer for shared refinement to facilitate holistic interaction among all constituent force, visual, and language tokens. This layer is composed of a multi-head self-attention mechanism with N_{heads} attention heads and a subsequent FFN, yielding $E_{enc} \in \mathbb{R}^{D_{model}}$. Subsequently, E_{enc} is channeled into a sparse Mixture-of-Experts layer. This layer employs $E = 4$ distinct expert networks, each realized as an independent MLP. A dynamic gating network determines the routing, selecting the most appropriate single expert (top $k = 1$) for each token in E_{enc} based on learned dispatch weights. The output from the MoE computation is then integrated back with the input to the MoE layer via a residual connection, yielding E_{fused}. The resulting sequence of fused multimodal features is passed through a final linear projection layer to match the dimensionality of the action expert.

Injecting Fused Features into the Action Flow Head. The sequence of fused multimodal features produced by the FVLMoE module serves as a guidance signal for the action generation process, which is formulated as a flow-based denoising model. This guidance is materialized by first extracting a specific sub-sequence, $G_{\text{FVLMoE}} \in \mathbb{R}^{H_{\text{action}} \times D_{\text{a}}}$, comprising the final H_{action} tokens from E_{FVLMoE}; these tokens encapsulate the most pertinent fused guidance for each step in the H_{action}-length action plan. G_{FVLMoE} is then combined via element-wise addition with $S_{\text{suffix}} \in \mathbb{R}^{H_{\text{action}} \times D_{\text{a}}}$ obtained from the primary VLM's processing of the current proprioceptive robot state $s_t \in \mathbb{R}^{D_s}$ and the noisy



(a) Insert USB (b) Pump Bottle (c) Insert Plug (d) Peel Cucumber (e) Wipe Board

Figure 4: Overview of task setups used in evaluation. (a) Insert USB, (b) pump bottle, (c) insert plug, (d) peel cucumber, and (e) wipe board. These tasks span diverse contact dynamics and manipulation skills, from precise insertions to tool-mediated surface interactions.

action trajectory $a_t^\tau \in \mathbb{R}^{H_{\text{action}} \times D_a}$ at denoising step τ, where D_s and D_a are the dimensionalities of the state and action spaces, respectively. This additive injection mechanism ensures that the rich, contact-aware contextual understanding developed by the FVLMoE directly modulates and refines the generated action sequence at each step of the predicted trajectory.

4.3 Datasets

To train ForceVLA, we curated a new dataset specifically focused on contact-rich manipulation tasks, emphasizing the synchronized capture of visual, proprioceptive, and force-torque data. Existing datasets often lack the comprehensive force interactions or the diversity of contact-driven scenarios necessary to develop robust force-aware policies.

Our data collection was performed using a Flexiv Rizon 7-DOF robotic arm equipped with a Dahuan adaptive gripper. Visual data was captured from two RGB-D cameras: one static third-person view (RealSense D435 at 1280x720, 30 FPS) and one wrist-mounted camera (RealSense D415 at 640x480, 30 FPS) providing egocentric perspectives. Data was collected via human teleoperation using a Quest3 VR interface with custom mappings to robot end-effector control. Five expert operators performed a total of 5 distinct contact-rich tasks: *bottle pumping, plug insertion, USB drive insertion, whiteboard wiping,* and *cucumber peeling*, as described in Section 5.1. For each task, operators were instructed to complete the objective while ensuring diverse and successful interaction patterns. We varied object positions and orientations across demonstrations to enhance data diversity.

The resulting dataset, which we term *ForceVLA-Data*, comprises a total of 244 trajectories, amounting to 140 thousand synchronized timesteps. All sensor streams were synchronized based on timestamps. Images were resized to 480x640 pixels and normalized. Actions were represented as target TCP pose and gripper width. The *ForceVLA-Data* dataset, along with our data collection code and processing scripts, will be made publicly available at website to facilitate further research in learning force-aware manipulation policies.

5 Experiments

This section presents a comprehensive suite of real-world contact-rich manipulation experiments and analytical studies to empirically validate the ForceVLA model. The evaluation is structured around four core research questions: (1) the overall effectiveness of ForceVLA compared to baselines that incorporate force without our specialized fusion mechanism; (2) the model's ability to generalize across unseen object instances, environmental variations, and task conditions; (3) the efficacy of the proposed FVLMoE architecture in achieving optimal multimodal fusion for contact-rich manipulation; and (4) the ability of the Mixture-of-Experts module to appropriately process heterogeneous input modalities and learn meaningful routing behaviors across expert networks.

5.1 Experimental Setups

To evaluate the effectiveness of ForceVLA, we conducted experiments on five diverse contact-rich manipulation tasks: Bottle Pumping, Plug Insertion, USB Drive Insertion, Whiteboard Wiping, and Cucumber Peeling, as in Figure 4. These tasks were chosen to assess fine-grained control, adaptability to varied initial conditions, and the utility of multimodal feedback, particularly force sensing. Each task introduces unique physical challenges: Bottle Pumping requires precise vertical pressing; Plug

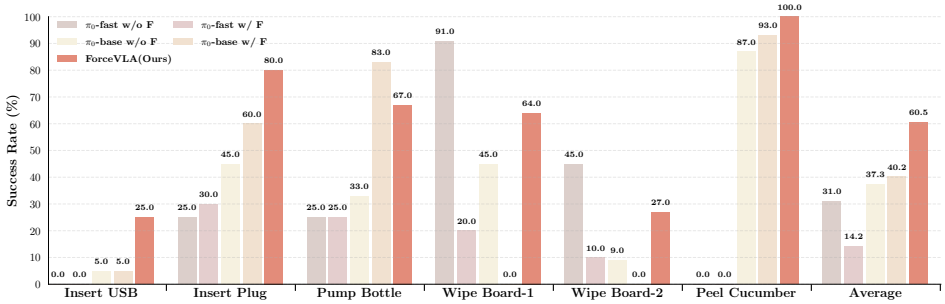


Figure 5: Main task success rates across different methods. ForceVLA significantly outperforms all baselines on five contact-rich tasks. Incorporating external force feedback improves performance for π_0-base model, while our method achieves the highest average success rate, demonstrating robust performance under complex interaction dynamics. "Wipe Board-1" indicates the success rate of successfully performing the wiping motion, while "Wipe Board-2" refers to the success rate of completely erasing the markings.

and USB Drive Insertions involve accurate alignment and force-controlled insertion; Whiteboard Wiping demands smooth trajectory control and surface contact; and Cucumber Peeling tests the ability to apply and maintain controlled force during continuous surface interaction.

We trained ForceVLA using approximately 50 expert demonstrations per task. Evaluation was conducted over 20 trials each for the insertion and pumping tasks, 10 trials for the more time-consuming whiteboard task, and 15 trials for the cucumber peeling task, each involving 15 peeling strokes. Success was defined by task-specific criteria such as complete insertion, effective wiping motion, or substantial cumulative peel coverage. These tasks were designed to rigorously probe ForceVLA's capacity to model and control complex, uncertain dynamics through the integration of vision, language, and force modalities.

Evaluation Metrics and Baselines. Model performance is primarily evaluated using the task success rate across all five challenging contact-rich manipulation tasks. For specific task like cucumber peeling, average peel length and minimum peeling times are also reported to provide a more nuanced assessment. To contextualize the performance of our proposed ForceVLA model, we compare it against several carefully selected baselines derived from the state-of-the-art π_0[10] architecture, which serves as our foundational model. The specific variants include π_0-base[10] w/o F (standard π_0 without force input), π_0-base[10] w/ F (π_0 with force signals directly concatenated to state inputs), and corresponding π_0-fast[25] configurations (w/o F and w/ F), representing potentially faster alternatives. The selection of π_0-base[10] allows comparison with a strong existing VLA method, while the 'inputForce' variants are crucial for demonstrating the efficacy of our FVLMoE fusion strategy over simpler force integration approaches.

5.2 Main Results

Overall Performance. As demonstrated in Figure 5, ForceVLA achieves an average success rate of 60.5% across all five tasks, significantly outperforming all baseline configurations. Compared to the standard π_0-base model without force feedback (π_0-base w/ F), which achieved an average of 37.3%, ForceVLA shows an improvement of 23.2%. This highlights the substantial benefit of incorporating and effectively processing multimodal information, including force. Table 1 further highlights ForceVLA's superior performance on the intricate *cucumber peeling* task. Our model excelled on both key metrics: it achieved the longest average peel length per stroke (14.12 cm ↑), indicating better ability to execute high-fidelity surface manipulation through stable tool orientation, adaptive contouring, and sustained surface contact compared to both π_0-base w/ F (13.17 cm) and π_0-base w/o F (10.27 cm). Concurrently, ForceVLA demonstrated superior overall efficiency by requiring the minimum number of strokes (7 ↓) to achieve a substantially peeled cucumber, significantly fewer than the 10 and 14 strokes needed by π_0-base w/ F and π_0-base w/o F, respectively. These combined results underscore ForceVLA's proficiency in maintaining consistent, effective tool-surface

Table 1: Performance of cucumber peeling.

Model	Avg. Peel Length (cm) ↑	Min. Strokes to Clean ↓
π_0-base[10] w/o F	10.27	14
π_0-base[10] w/ F	13.17	10
ForceVLA (Ours)	**14.12**	**7**

Table 2: Success rates (%) of different models under various experimental conditions. Maximum values in each column are highlighted in **bold**; second-best values are underlined.

Model	Object Gen. 1	Object Gen. 2	Height Gen.	Visual Occlusion	Unstable Socket	Average
π_0-base[10] w/o F	48.00%	10.00%	66.67%	60.00%	10.00%	38.93%
π_0-base[10] w/ F	32.00%	10.00%	77.78%	30.00%	10.00%	31.96%
π_0-fast[25] w/o F	**80.00%**	35.00%	**88.89%**	50.00%	10.00%	52.78%
π_0-fast[25] w/ F	32.00%	5.00%	44.44%	50.00%	**30.00%**	32.29%
ForceVLA (Ours)	**80.00%**	**40.00%**	**88.89%**	**90.00%**	20.00%	**63.78%**

interaction and executing efficient, goal-directed motions in tasks demanding continuous and precise force modulation.

Effectiveness of Force Fusion via FVLMoE. Introducing raw force signals into π_0-base boosts performance from 37.3% to 40.2%, confirming the utility of force feedback. However, ForceVLA surpasses both with 60.5%, indicating that effective fusion—enabled by our FVLMoE module—is essential for fully leveraging tactile information. This demonstrates that beyond the presence of force data, how it is integrated is critical to performance gains.

Selection of π_0-base and π_0-fast model. For our foundational baseline, we evaluated π_0-base and π_0-fast variants. The π_0-base architecture demonstrated superior overall performance: π_0-base w/ F (40.2%) and π_0-base w/o F (37.3%) significantly outperformed π_0-fast w/ F (14.2%) and π_0-fast w/o F (31.0%). While π_0-fast variants exhibited a comparative advantage solely on the whiteboard wiping task—potentially due to a simpler action generation mechanism being more attuned to such motions, the π_0-fast architecture's performance notably degraded (from 31.0% to 14.2%) when raw force input was directly added. We attribute this sensitivity to its highly optimized and compact token space, which is likely disrupted by naively projected force tokens lacking corresponding large-scale pre-training. Conversely, π_0-base modestly benefited from direct force input, with its larger representational capacity presumably allowing for partial utilization of these new sensory signals. Given its superior aggregate performance and more robust handling of naive force integration, π_0-base was selected as the primary baseline for developing and evaluating ForceVLA.

5.3 Model Generalization

To evaluate ForceVLA's generalization capabilities, we designed five experimental settings with increasing task variability and physical uncertainty, as illustrated in Figure 6. These settings include: (1) **Object Gen. 1**, which varies the bottle type in the bottle pumping task; (2) **Object Gen. 2**, which changes the plug type in the plug insertion task; (3) **Height Gen.**, which adjusts the initial bottle height and measures success under torque limits; (4) **Visual Occlusion**, where parts of the plug and socket are obscured; and (5) **Unstable Socket**, introducing physical instability via clutter beneath the socket. These variations test both perceptual robustness and physical adaptability, with results summarized in Table 2.

Across all settings, ForceVLA exhibited superior generalization, particularly in scenarios requiring fine physical interaction. In Object Gen. 1, it achieved an 80.00% success rate, outperforming baselines that lacked force input or processed it naively. In the Height Gen. setting, ForceVLA effectively scaled its interaction forces to variable depths, avoiding torque limit violations seen in other models. Furthermore, ForceVLA maintained high success under visual degradation (90.00% in Visual Occlusion), reflecting its reliance on multimodal feedback beyond visual cues. Notably, while π_0-fast w/f performed best in the Unstable Socket task, this advantage stems from its frequent pausing behavior—when the end-effector pauses while in contact with the plug, the socket is less likely to wobble, effectively reducing the task difficulty. However, this stuttering limits its inference speed and reactiveness to real-time force changes, making it less competitive for general contact-rich manipulation tasks requiring prompt trajectory adjustments. These results underscore the critical role of the proposed FVLMoE architecture in intelligently integrating force information—not just for sensing contact, but for modulating action in response to dynamic physical conditions—enabling more versatile and robust robotic manipulation.

5.4 Ablation Studies

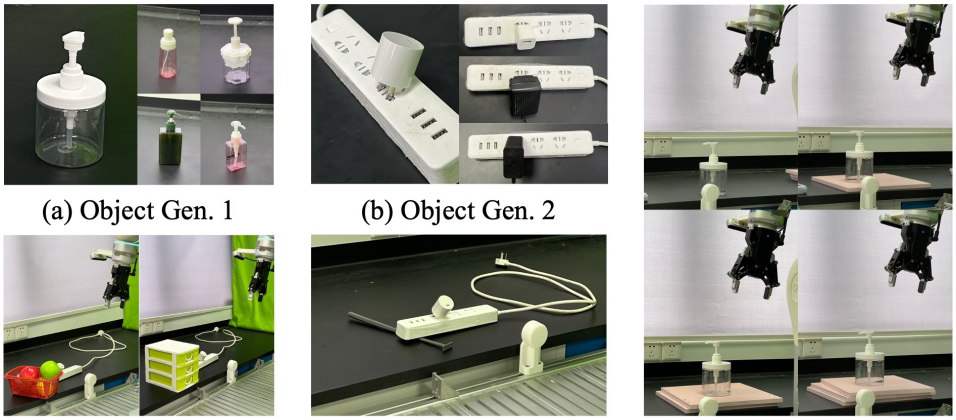


Figure 6: Variants of generalization settings used in our experiments. (a–b) Different object geometries; (c) variation in socket height; (d) partial visual occlusion; (e) unstable socket conditions. These scenarios evaluate robustness under diverse physical and perceptual perturbations.

To validate the architectural design of *ForceVLA*, particularly the integration of force feedback, we conducted comprehensive ablation studies shown in Table 3. We compared early, late, and our proposed fusion strategy. Early fusion methods, such as "linear before VLM" and "MoE before VLM," which inject force data prior to the visual-language model (VLM), significantly degraded performance. Notably, the MoE-based early fusion failed entirely (0% success rate), highlighting that altering the input representations of a pretrained VLM disrupts its learned feature distributions and undermines its capacity to process visual-linguistic signals effectively.

Table 3: Ablation Results

Model	Success Rate
baseline[10]	45%
linear before VLM	55%
MoE before VLM	0
concate after VLM	60%
ForceVLA (Ours)	**80%**

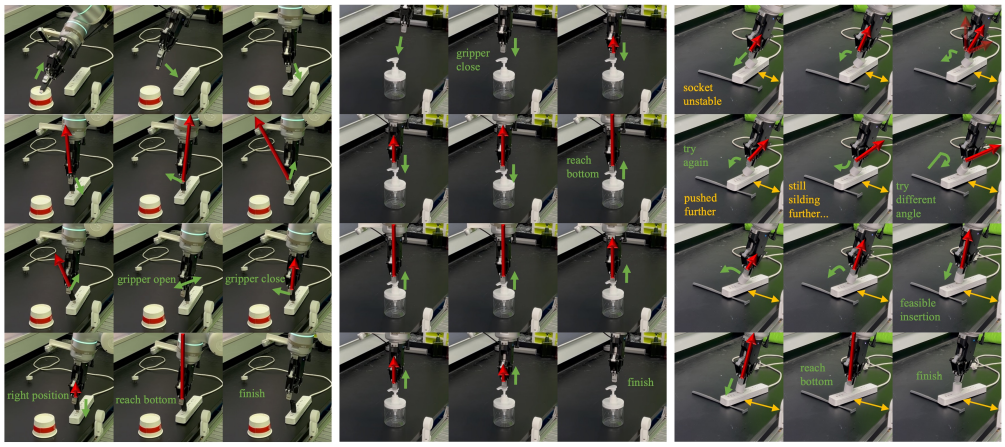
Late fusion strategies fared better. The "concatenate after VLM" method, equivalent to a basic baseline where force features are appended at the decoding stage, improved success to 60%—demonstrating the utility of force sensing. However, our proposed **ForceVLA** architecture achieved a markedly higher 80% success rate. By introducing force features after the VLM's core encoding and using the FVLMoE module for adaptive fusion, ForceVLA enables specialized routing and deeper multimodal interaction. These results confirm two core design insights: force should be introduced post-VLM to preserve pretrained representations, and sophisticated fusion (via FVLMoE) is essential to fully leverage force in guiding contact-rich robotic behavior.

5.5 Visualization and Case Studies

Figure 7 illustrates ForceVLA's ability to adapt motion in response to contact feedback during complex manipulation tasks. In the USB insertion task, when initial attempts failed due to misalignment, ForceVLA re-oriented or re-grasped the drive to achieve successful insertion—behaviors absent in baseline models, which repeated failed motions or applied excessive force. Similarly, in the "Unstable Socket" scenario (Figure 7c), ForceVLA maintained compliant control as the socket shifted, dynamically adjusting the plug's pose to complete insertion, while baselines lost tracking and failed. These examples highlight a key insight: simply adding force input does not ensure closed-loop adaptation. ForceVLA's FVLMoE module enables deep fusion of force, vision, and language, supporting precise, context-aware control and robust generalization under dynamic physical conditions.

6 Conclusion

In conclusion, we propose ForceVLA, a framework that bridges the gap between high-level modality (vision/language) and low-level physical sensing (force) for contact-rich manipulation. At its core, ForceVLA introduces **FVLMoE**, a Mixture-of-Experts module that dynamically fuses visual, linguistic, and force modalities to enable fine-grained, context-aware control. Our experiments across five



(a) Insert USB (b) Pump Bottle (c) Insert Plug

Figure 7: Trajectory visualizations across tasks and conditions. (a) USB insertion, (b) bottle pumping, and (c) plug insertion under stable and unstable socket conditions. Each sequence illustrates how ForceVLA adapts its actions in response to contact dynamics, retrying or adjusting pose when failures occur, ultimately achieving successful task completion.

challenging tasks show that ForceVLA significantly outperforms strong π_0-based baselines, achieving an average success rate improvement of 23.2% and up to 80% success on individual tasks. Ablation studies further validate the benefits of late-stage force fusion and expert routing. We also contribute **ForceVLA-Data**, a new dataset for multimodal contact-rich manipulation. Through the co-design of our architectural approach and dataset, we demonstrate meaningful progress toward VLA systems that exhibit greater adaptive behavior and physical intelligence in manipulation scenarios.

Limitation. Firstly, ForceVLA currently utilizes estimated external wrench values. While this approach has proven effective, these estimations may not always capture the full precision afforded by direct high-fidelity measurements, particularly in scenarios demanding extreme haptic sensitivity. Potential enhancements include exploring the integration of superior sensors or advanced calibration techniques to further refine fine-grained control capabilities. Secondly, ForceVLA's experimental validation was predominantly conducted on robotic platforms with integrated, and typically high-cost, force-torque sensing, which can naturally limit broader accessibility. To promote wider practical deployment and help democratize force-aware manipulation research, we are actively assessing ForceVLA's adaptability and performance on more common, lower-cost platforms equipped with external or retrofitted force sensors. Thirdly, our work focuses exclusively on real-world data collection and evaluation to ensure findings are directly applicable to physical deployment scenarios, as multi-body contact dynamics are central to our problem. While this real-world-first approach addresses the sim-to-real gap, developing high-fidelity simulation environments for our diverse contact-rich tasks was beyond the scope of current resources. We recognize that large-scale training with simulated data represents a promising avenue for expanding the range of tasks our model can address and view this as an important direction for future work.

Acknowledgment

This work was supported in part by the National Key R&D Program of China (Grant No.2024YFB4707600), National Natural Science Foundation of China (No.62576109, 62072112), National Natural Science Foundation of China under Grant 52505029, Natural Science Foundation of Shanghai under Grant 25ZR1401191, Science and Technology Major Project of Jiangsu Province (No.BG2024041), the Shanghai Commitee of Science and Technology, China(Grant No.24511103200) by the National Key Research and Development Project of China (No.2022ZD0160102), Shanghai Artificial Intelligence Laboratory, XPLORER PRIZE grants.

References

[1] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

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

[3] Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning*, pages 1723–1736. PMLR, 2023.

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

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

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

[7] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[8] Wenwen Zhuang, Xin Huang, Xiantao Zhang, and Jin Zeng. Math-puma: Progressive upward multimodal alignment to enhance mathematical reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 26183–26191, 2025.

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

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

[11] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024.

[12] Xiang Zhang, Changhao Wang, Lingfeng Sun, Zheng Wu, Xinghao Zhu, and Masayoshi Tomizuka. Efficient sim-to-real transfer of contact-rich manipulation skills with online admittance residual learning. In *Conference on Robot Learning*, pages 1621–1639. PMLR, 2023.

[13] Michael Noseworthy, Bingjie Tang, Bowen Wen, Ankur Handa, Chad Kessens, Nicholas Roy, Dieter Fox, Fabio Ramos, Yashraj Narang, and Iretiayo Akinola. Forge: Force-guided exploration for robust contact-rich manipulation under uncertainty. *IEEE Robotics and Automation Letters*, 2025.

[14] Sung Soo Kim, Manuel Gomez-Ramirez, Pramodsingh H Thakur, and Steven S Hsiao. Multimodal interactions between proprioceptive and cutaneous signals in primary somatosensory cortex. *Neuron*, 86(2):555–566, 2015.

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

[16] Wei Zhao, Pengxiang Ding, Min Zhang, Zhefei Gong, Shuanghao Bai, Han Zhao, and Donglin Wang. Vlas: Vision-language-action model with speech instructions for customized robot manipulation. *arXiv preprint arXiv:2502.13508*, 2025.

[17] Jiaming Liu, Mengzhen Liu, Zhenyu Wang, Lily Lee, Kaichen Zhou, Pengju An, Senqiao Yang, Renrui Zhang, Yandong Guo, and Shanghang Zhang. Robomamba: Multimodal state space model for efficient robot reasoning and manipulation. *arXiv preprint arXiv:2406.04339*, 2024.

[18] Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. Vima: General robot manipulation with multimodal prompts. *arXiv preprint arXiv:2210.03094*, 2(3):6, 2022.

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

[20] Cunxin Fan, Xiaosong Jia, Yihang Sun, Yixiao Wang, Jianglan Wei, Ziyang Gong, Xiangyu Zhao, Masayoshi Tomizuka, Xue Yang, Junchi Yan, et al. Interleave-vla: Enhancing robot manipulation with interleaved image-text instructions. *arXiv preprint arXiv:2505.02152*, 2025.

[21] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_0.5$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

[22] Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. *arXiv preprint arXiv:2503.22020*, 2025.

[23] Junjie Wen, Yichen Zhu, Jinming Li, Zhibin Tang, Chaomin Shen, and Feifei Feng. Dexvla: Vision-language model with plug-in diffusion expert for general robot control. *arXiv preprint arXiv:2502.05855*, 2025.

[24] Qingwen Bu, Hongyang Li, Li Chen, Jisong Cai, Jia Zeng, Heming Cui, Maoqing Yao, and Yu Qiao. Towards synergistic, generalized, and efficient dual-system for robotic manipulation. *arXiv preprint arXiv:2410.08001*, 2024.

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

[26] Chengmeng Li, Junjie Wen, Yan Peng, Yaxin Peng, Feifei Feng, and Yichen Zhu. Pointvla: Injecting the 3d world into vision-language-action models. *arXiv preprint arXiv:2503.07511*, 2025.

[27] 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*, page 02783649241273668, 2023.

[28] Yanjiang Guo, Yucheng Hu, Jianke Zhang, Yen-Jen Wang, Xiaoyu Chen, Chaochao Lu, and Jianyu Chen. Prediction with action: Visual policy learning via joint denoising process. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[29] Zhongyi Zhou, Yichen Zhu, Minjie Zhu, Junjie Wen, Ning Liu, Zhiyuan Xu, Weibin Meng, Ran Cheng, Yaxin Peng, Chaomin Shen, et al. Chatvla: Unified multimodal understanding and robot control with vision-language-action model. *arXiv preprint arXiv:2502.14420*, 2025.

[30] Jiaming Liu, Hao Chen, Pengju An, Zhuoyang Liu, Renrui Zhang, Chenyang Gu, Xiaoqi Li, Ziyu Guo, Sixiang Chen, Mengzhen Liu, et al. Hybridvla: Collaborative diffusion and autoregression in a unified vision-language-action model. *arXiv preprint arXiv:2503.10631*, 2025.

[31] Yansong Wu, Zongxie Chen, Fan Wu, Lingyun Chen, Liding Zhang, Zhenshan Bing, Abdalla Swikir, Sami Haddadin, and Alois Knoll. Tacdiffusion: Force-domain diffusion policy for precise tactile manipulation. *arXiv preprint arXiv:2409.11047*, 2024.

[32] Yifan Hou, Zeyi Liu, Cheng Chi, Eric Cousineau, Naveen Kuppuswamy, Siyuan Feng, Benjamin Burchfiel, and Shuran Song. Adaptive compliance policy: Learning approximate compliance for diffusion guided control. *arXiv preprint arXiv:2410.09309*, 2024.

[33] Wenhai Liu, Junbo Wang, Yiming Wang, Weiming Wang, and Cewu Lu. Forcemimic: Force-centric imitation learning with force-motion capture system for contact-rich manipulation. *arXiv preprint arXiv:2410.07554*, 2024.

[34] Zihao He, Hongjie Fang, Jingjing Chen, Hao-Shu Fang, and Cewu Lu Foar. Force-aware reactive policy for contact-rich robotic manipulation. *arXiv preprint arXiv:2411.15753*, 2024.

[35] William Xie and Nikolaus Correll. Towards forceful robotic foundation models: a literature survey. *arXiv preprint arXiv:2504.11827*, 2025.

[36] Peng Hao, Chaofan Zhang, Dingzhe Li, Xiaoge Cao, Xiaoshuai Hao, Shaowei Cui, and Shuo Wang. Tla: Tactile-language-action model for contact-rich manipulation. *arXiv preprint arXiv:2503.08548*, 2025.

[37] Zihang Zhao, Yuyang Li, Wanlin Li, Zhenghao Qi, Lecheng Ruan, Yixin Zhu, and Kaspar Althoefer. Tac-man: Tactile-informed prior-free manipulation of articulated objects. *IEEE Transactions on Robotics*, 2024.

[38] Chaofan Zhang, Peng Hao, Xiaoge Cao, Xiaoshuai Hao, Shaowei Cui, and Shuo Wang. Vtla: Vision-tactile-language-action model with preference learning for insertion manipulation. *arXiv preprint arXiv:2505.09577*, 2025.

[39] Yiyang Ling, Karan Owalekar, Oluwatobiloba Adesanya, Erdem Bıyık, and Daniel Seita. Impact: Intelligent motion planning with acceptable contact trajectories via vision-language models. *arXiv preprint arXiv:2503.10110*, 2025.

[40] Huaijiang Zhu, Tong Zhao, Xinpei Ni, Jiuguang Wang, Kuan Fang, Ludovic Righetti, and Tao Pang. Should we learn contact-rich manipulation policies from sampling-based planners? *IEEE Robotics and Automation Letters*, 2025.

[41] Joshua Jones, Oier Mees, Carmelo Sferrazza, Kyle Stachowicz, Pieter Abbeel, and Sergey Levine. Beyond sight: Finetuning generalist robot policies with heterogeneous sensors via language grounding. *arXiv preprint arXiv:2501.04693*, 2025.

[42] Hao-Shu Fang, Hongjie Fang, Zhenyu Tang, Jirong Liu, Chenxi Wang, Junbo Wang, Haoyi Zhu, and Cewu Lu. Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot. *arXiv preprint arXiv:2307.00595*, 2023.

[43] Yashraj Narang, Kier Storey, Iretiayo Akinola, Miles Macklin, Philipp Reist, Lukasz Wawrzyniak, Yunrong Guo, Adam Moravanszky, Gavriel State, Michelle Lu, et al. Factory: Fast contact for robotic assembly. *arXiv preprint arXiv:2205.03532*, 2022.

[44] Bingjie Tang, Michael A Lin, Iretiayo Akinola, Ankur Handa, Gaurav S Sukhatme, Fabio Ramos, Dieter Fox, and Yashraj Narang. Industreal: Transferring contact-rich assembly tasks from simulation to reality. *arXiv preprint arXiv:2305.17110*, 2023.

[45] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

[46] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

[47] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020.

[48] Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. Glam: Efficient scaling of language models with mixture-of-experts. In *International conference on machine learning*, pages 5547–5569. PMLR, 2022.

[49] Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. *Advances in Neural Information Processing Systems*, 34:8583–8595, 2021.

[50] Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. St-moe: Designing stable and transferable sparse expert models. *arXiv preprint arXiv:2202.08906*, 2022.

[51] Basil Mustafa, Carlos Riquelme, Joan Puigcerver, Rodolphe Jenatton, and Neil Houlsby. Multimodal contrastive learning with limoe: the language-image mixture of experts. *Advances in Neural Information Processing Systems*, 35:9564–9576, 2022.

[52] Han Zhao, Wenxuan Song, Donglin Wang, Xinyang Tong, Pengxiang Ding, Xuelian Cheng, and Zongyuan Ge. More: Unlocking scalability in reinforcement learning for quadruped vision-language-action models. *arXiv preprint arXiv:2503.08007*, 2025.

[53] Qiang Liu. Rectified flow: A marginal preserving approach to optimal transport. *arXiv preprint arXiv:2209.14577*, 2022.

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

[55] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023.

[56] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1930–1939, 2018.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The answer NA means that the abstract and introduction do not include the claims made in the paper

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: The paper clearly states the limitation of this work.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: No theoretical result.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper includes details for experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: This paper includes details for the experiments.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: This paper includes details for the experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The results compare with baselines and ablation studies.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: This paper provides the training details and required computation resources.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This paper follows the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: This paper is not related to societal impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper does not have the risk of misuse language models.

    Guidelines:

    - The answer NA means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: This paper is not related to assets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: This paper includes new datasets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper is not related to human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

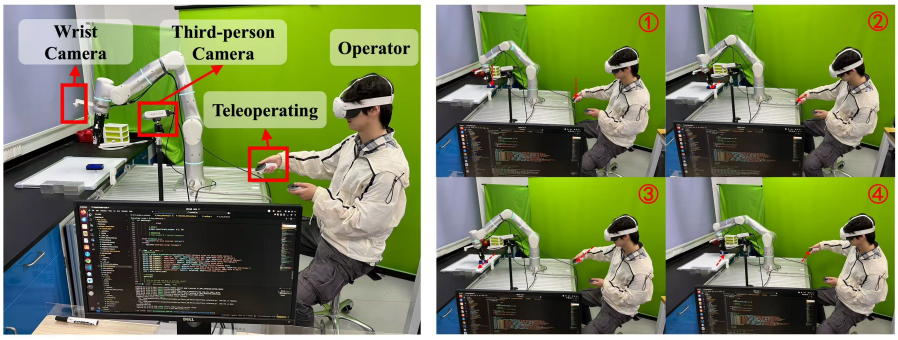
    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Data Collection System



(a) Data Collection with Teleoperation Setup (b) Teleoperation Demonstration

Figure 8: Our data collection system setup.

Our data collection system is depicted in Figure 8(a). The setup features a robotic arm equipped with both a wrist-mounted camera and a static third-person view camera to capture diverse visual perspectives. An operator, wearing a Quest 3 headset and using hand-held controllers, teleoperates the robot arm. A nearby computer runs the necessary software for data acquisition, which includes programs for interfacing with the robot, synchronizing sensor streams, and managing communication with the VR teleoperation hardware. Figure 8(b) illustrates the operator teleoperating the robotic arm to collect demonstration data for the *wipe board* task.

B Training Details

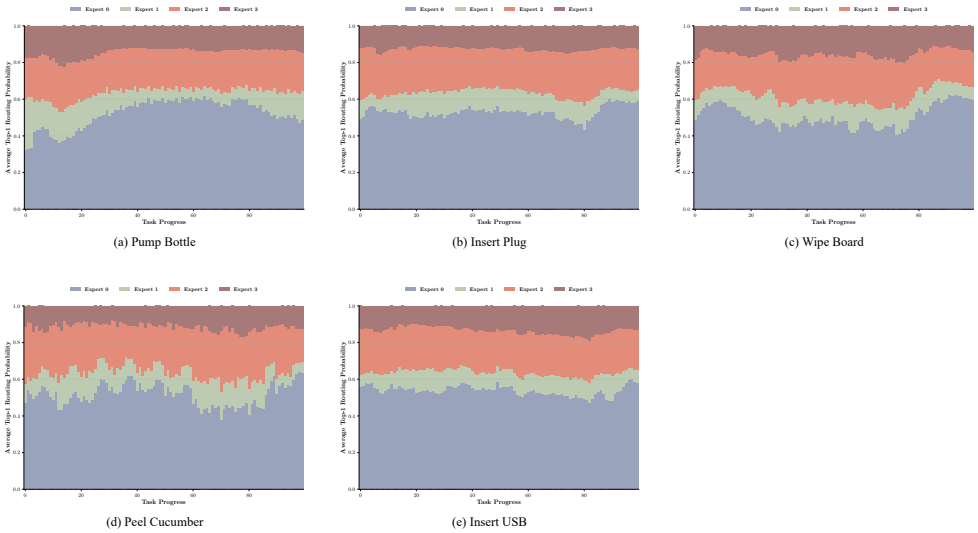
The models were mainly trained on compute nodes equipped with 8× NVIDIA RTX 4090 GPUs (24 GB VRAM each), 64 physical CPU cores, and 251 GB system RAM, using Adam optimization ($\beta_1 = 0.9$, $\beta_2 = 0.95$) with a peak learning rate of 2.5×10^{-5} decaying to 2.5×10^{-6} over 30,000 steps. Multi-task training utilized data parallelism across 2 GPUs (global batch size 16, effective 2048 via gradient accumulation) as additional GPUs provided diminishing returns due to communication overhead, completing 30,000 steps in ∼12 hours, while single-task training used 1 GPU for 10,000 steps (∼9 hours), both employing `bfloat16` precision with gradient clipping ($\|\nabla\| = 1.0$).

The dimensionalities and key parameters of ForceVLA's core processing modules: Input Projections, the FVLMoE block, and the Action Output Head are detailed in Table 4.

Table 4: Focused view of ForceVLA's key architectural components: Input Projections, FVLMoE, and Action Output. Dimensions are indicative (e.g., D_{VLM}, $D_{\text{act\_e}}$ for VLM and Action Expert).

Layer	**Key Parameters / Dimensions**
Input Projections	
Force Projection	Linear; Input: 6 (F/T), Output: $D_{\text{VLM}} = 2048$
State Projection	Linear; Input: $D_{\text{state}} = 32$, Output: $D_{\text{act\_e}} = 1024$
Action Projection	Linear; Input: $D_{\text{action}} = 32$, Output: $D_{\text{act\_e}} = 1024$
Action-Time MLP	2-layer MLP; Input: $2 \times D_{\text{act\_e}}$, Hidden/Output: $D_{\text{act\_e}}$; Swish activation
FVLMoE Module	
Input	Concatenation: $N_{\text{VL}} \times D_{\text{VLM}}$ (V-L features) & $1 \times D_{\text{VLM}}$ (Force token)
Pre-MoE Encoder	Transformer Encoder Block; $D_{\text{model}} = 2048$, $N_{\text{H}} = 8$, $D_{\text{h}} = 256$; MLP (expansion factor 1)
MoE Layer	Sparse MoE; $E = 4$ experts (MLPs: $D_{\text{model}} \to D_{\text{model}}$), Top-$k = 1$; Router: $D_{\text{model}} \to E$
Output Projection	Linear; Input: $D_{\text{model}} = 2048$, Output: $D_{\text{act\_e}} = 1024$
Action Output Head	
Action Output Projection	Linear; Input: $D_{\text{act\_e}} = 1024$, Output: $D_{\text{action}} = 32$

C Router Analysis



(a) Pump Bottle (b) Insert Plug (c) Wipe Board

(d) Peel Cucumber (e) Insert USB

Figure 9: Open-loop evaluation of expert load across different task completion percentages for various tasks: (a) Pump Bottle, (b) Insert Plug, (c) Wipe Board, (d) Peel Cucumber, and (e) Insert USB. Each subplot represents the average expert load (vertical axis) as a function of the task completion percentage (horizontal axis) over the episodes in the test dataset.

To analyze routing dynamics, we first measured the probability distribution over expert selections for each token as it was processed by the router in the MoE module. For variable-length episodes, we applied percentile-based normalization: each task ($\sim$10 episodes) was processed by segmenting every episode's token sequence into 100 temporally equidistant intervals, computing the mean top-1 probability per segment, and then averaging these means across episodes. This ensured cross-episode comparability while preserving temporal routing dynamics.

As shown in Figure 9, different tasks exhibit distinct expert utilization patterns. Some tasks (e.g., *insert plug*, *peel cucumber*) show clear temporal specialization, where certain experts dominate specific phases of the task. Others (e.g., *wipe board*) demonstrate a more consistent preference for a single expert throughout execution. These trends suggest that the router learns to allocate computation dynamically across experts based on task-specific semantics and temporal structure.

What's more, we found that Expert 0 dominates nearly half of the tokens across multiple tasks. This persistent activation suggests that Expert 0 may function as a general-purpose expert, responsible for the fusion of multiple modalities or routine control primitives that are shared across tasks. Its broad involvement contrasts with the more selective, phase-specific activation of Expert 1 or Expert 3, reinforcing the hypothesis of functional specialization among experts. Such asymmetry in routing frequency reflects not only temporal semantic variance within tasks but also architectural bias toward certain experts, potentially shaped during training.

D Multi-task Evaluation

Table 5: Multi-task joint training success rates (%). ForceVLA (Ours) demonstrates superior average performance and excels or matches the best performance in all individual tasks. Best performance(s) in each column are in **bold**.

Model	Pump Bottle	Insert Plug	Insert USB	Wipe Board	Average SR
π_0-fast w/o F	0.0%	0.0%	0.0%	0.0%	0.0%
π_0-fast w/ F	0.0%	0.0%	0.0%	0.0%	0.0%
π_0-base w/o F	20.0%	0.0%	0.0%	0.0%	5.0%
π_0-base w/ F	50.0%	**100.0%**	**10.0%**	10.0%	42.5%
ForceVLA (Ours)	**80.0%**	**100.0%**	**10.0%**	**80.0%**	**67.5%**

The results of joint multi-task learning are detailed in Table 5. Notably, both π_0-fast variants (0% average success rate) failed to acquire skills in this setting, indicating their limited capacity for diverse, concurrent learning. The π_0-base w/o F model also performed poorly (5% average success rate), managing only 20% success on a single task. Adding direct force input (π_0-base w/ F) improved the average performance to 42.5%, primarily due to its success in the *insert plug* task. Our **ForceVLA** model demonstrated superior multi-task capabilities, achieving the highest average success rate of 67.5%. It obtained 80% success in both *pump bottle* and *wipe board* tasks, and matched top performance in *insert plug* (100%) and *insert usb* (10%). This robust performance across multiple distinct tasks indicates ForceVLA's capacity for concurrent skill learning, proficient instruction following for varied goals, and highlights the role of its FVLMoE architecture in utilizing multimodal cues, particularly force, within a shared policy.

E Detailed Breakdown of Failure Modes

- **Plug Insertion**: w/o force models generally exhibit cruder behavior during insertion, unable to adjust their strategy based on interaction state changes. w/ force models can generally perceive changes in external forces but are not flexible enough in adjusting their trajectory. ForceVLA can better maintain continuous contact, follow the plug's surface, make fine adjustments to the force angle and duration within a more precise range, and terminate the action more promptly upon full insertion.
- **USB Insertion**: The overall success rate for this task is low. A common problem across all five models is the difficulty in aligning with the USB port, which we attribute to insufficient visual clarity and the backbone model's inability to process fine-grained visual information. However, ForceVLA's success rate is slightly higher because it exhibits clear autonomous adjustment or re-attempt behaviors upon feeling external force from contact with the socket.
- **Bottle Pumping**: Simply pressing a seen bottle could be completed with 100% success by the pi0-base models and ForceVLA. Therefore, during testing for this task, we introduced additional visual occlusions and background changes. Most of the failure modes occurred under these specific variations. w/o force models often missed the pump or did not press it fully. ForceVLA was more robust but sometimes pressed off-center.
- **Cucumber Peeling**: w/o force models were significantly more prone to breaking the peel mid-way, unable to peel continuously from end to end, and sometimes peeled too deeply, indicating a lack of control over the peeling force. w/ force models peeled more stably with a wider peel, but still had issues with breaking the peel and not following the cucumber's curvature well. ForceVLA could overcome these issues.
- **Board Wiping**: A common problem for all models was the inability to pick up the eraser, possibly because its placement was far from the base camera, resulting in low resolution, and its color was similar to the black table, making precise localization difficult. In the remaining trials, if a small part of the writing wasn't erased, we classify as a failure. Additionally, if a model didn't stop after 5 minutes, we counted it as a failure. The w/o force models started wiping in the air or pressed too hard, leaving scratches, due to their inability to perceive force. ForceVLA made closer contact with the board and applied a more appropriate wiping force than other models.

F Ablation Study: Force Input Masking

To directly assess the value of the force modality within our MoE architecture, we conducted an ablation study where the force input to the ForceVLA model was masked during inference. Table 6 presents the quantitative results comparing ForceVLA with and without force inputs across four contact-rich manipulation tasks. Note that the Peel Cucumber task was not included in this ablation due to time constraints.

As shown in Table 6, removing force inputs leads to substantial performance degradation in most tasks, with particularly severe drops in Plug Insertion (60%), Pump Bottle (37%), and USB Insertion (25%). The Wipe Board task shows minimal degradation (0.3%), likely due to its relatively simpler contact dynamics.

Table 6: Success rates comparison between ForceVLA and ForceVLA without force inputs. The drop column indicates the absolute percentage decrease in success rate when force information is removed.

Task	ForceVLA (%)	ForceVLA w/o Force (%)	Drop (%)
Plug Insertion	80.0	20.0	60.0
USB Insertion	25.0	0.0	25.0
Pump Bottle	67.0	30.0	37.0
Wipe Board	27.0	26.7	0.3

F.1 Failure Mode Analysis

Through detailed analysis of the failed attempts, we identified several characteristic failure modes when force information is unavailable:

Plug Insertion: The end-effector could only push a short distance in the insertion direction, with insufficient trajectory curvature to properly insert the plug into the socket. Alternatively, the end-effector would move to the plug's surface and perform an arcing push motion near the contact point without establishing actual contact, resembling an overfitted insertion trajectory.

USB Insertion: The gripper's closing width was excessively large, preventing it from securely grasping the USB drive. This behavior suggests overfitting to the gripper width values observed during training, resulting in an inability to achieve a firm grasp.

Pump Bottle: The policy repeatedly applied excessive force, causing the gripper to bend. Additionally, off-center pressing caused the bottle to spring away from the gripper, indicating poor force regulation.

Wipe Board: The number of wiping attempts was significantly reduced, resulting in most cases where the board surface was not adequately cleaned.

F.2 Discussion

We hypothesize that these failure modes arise from the following mechanism: Without force input, the FVLMoE module's increased parameter count makes the model more susceptible to overfitting. During inference, the policy exhibits stereotypical behaviors that are misguided by vision alone, reflecting an overfit to the action distribution of the training data rather than reactive responses to actual contact conditions.

Conversely, the fact that the full ForceVLA model with the additional force modality does not suffer from this overfitting strongly indicates that our architectural design effectively leverages force signals to regularize the policy. This regularization enables the model to achieve robust, reactive behaviors instead of merely memorizing training trajectories, thereby validating the importance of force feedback in our multimodal MoE framework.

G Real-world Experiments Visualization

In this section, we present key frames from real-world experiment videos. Each visualization contrasts failure cases of baseline models with successful task completions by our ForceVLA model under similar conditions.

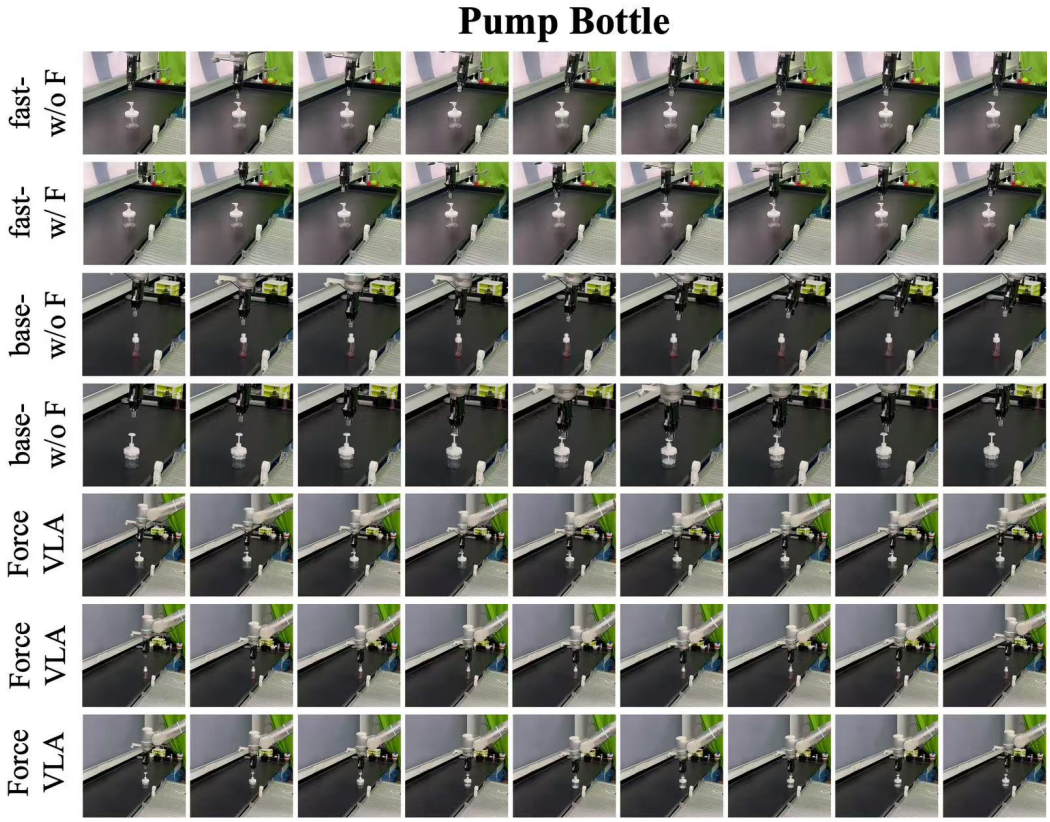


Figure 10: Key frames from Pump Bottle task videos.

Figure 11: Key frames from Insert USB task videos.

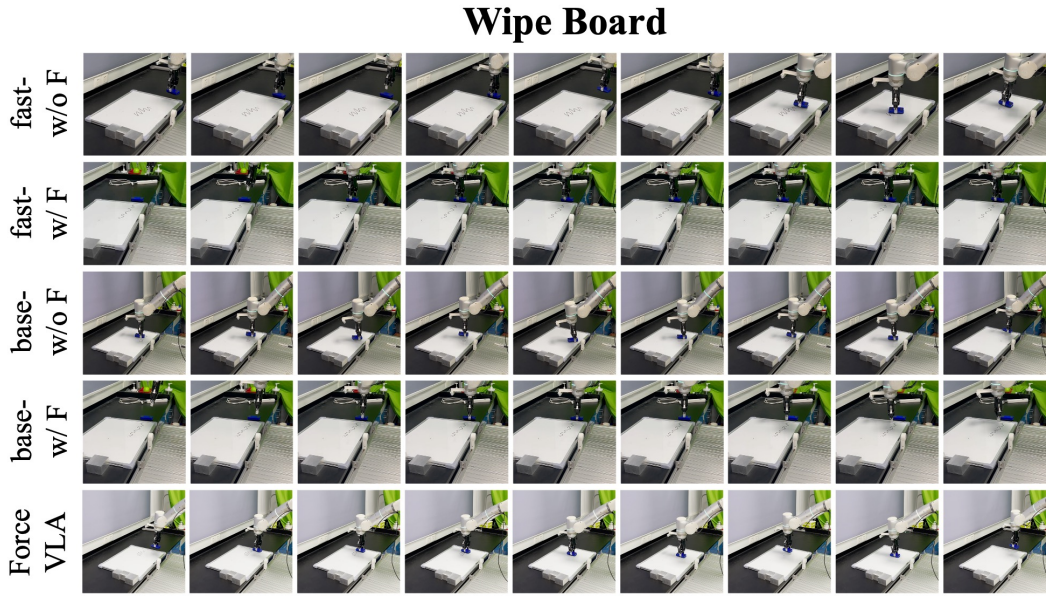


Figure 12: Key frames from Wipe Board task videos.

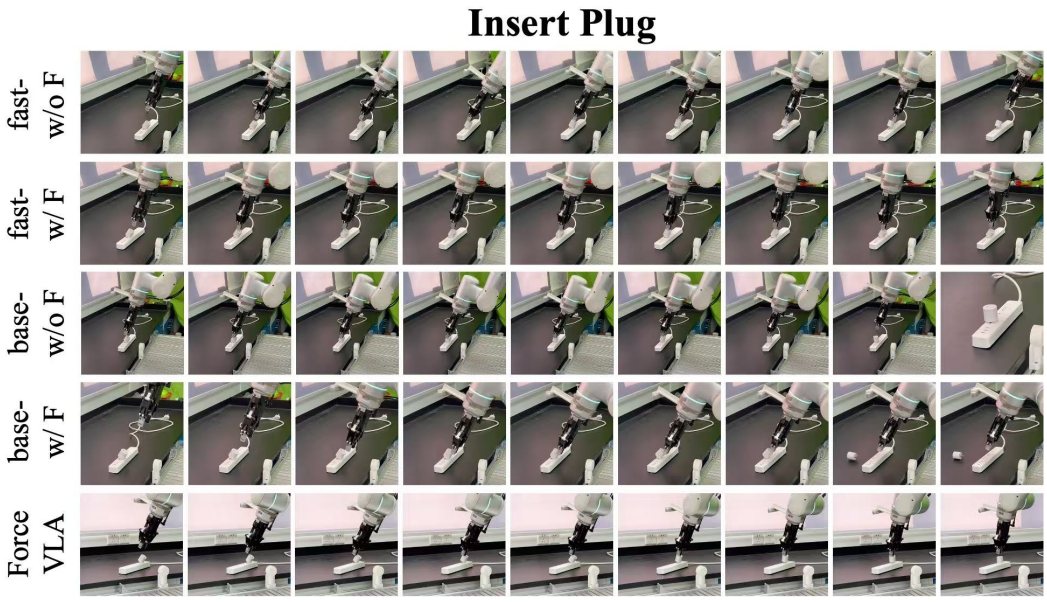


Figure 13: Key frames from Insert Plug task videos.

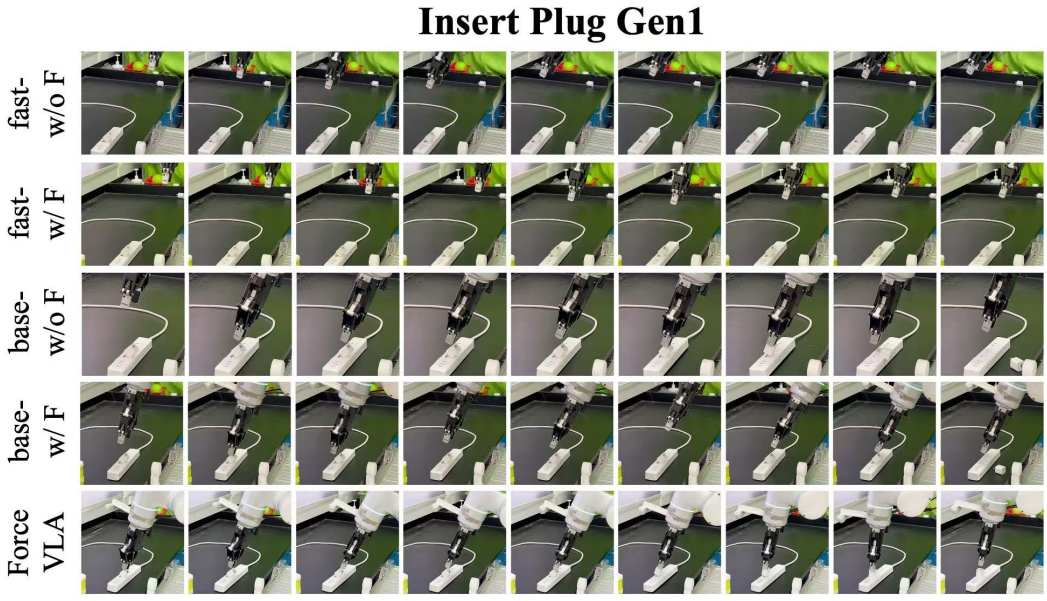


Figure 14: Key frames from Insert Plug Generalization task 1 videos.

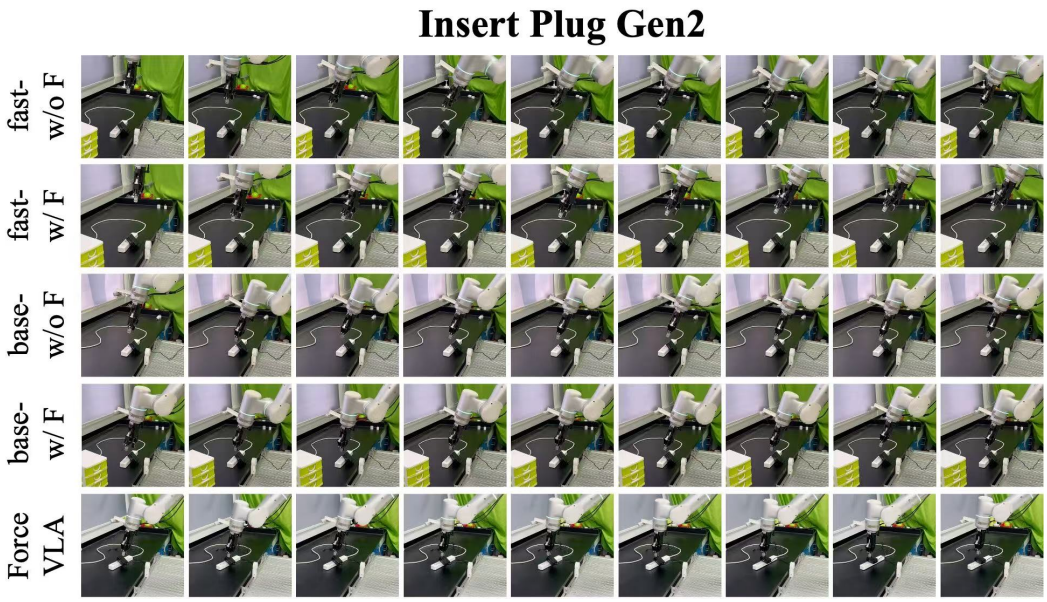


Figure 15: Key frames from Insert Plug Generalization task 2 videos.

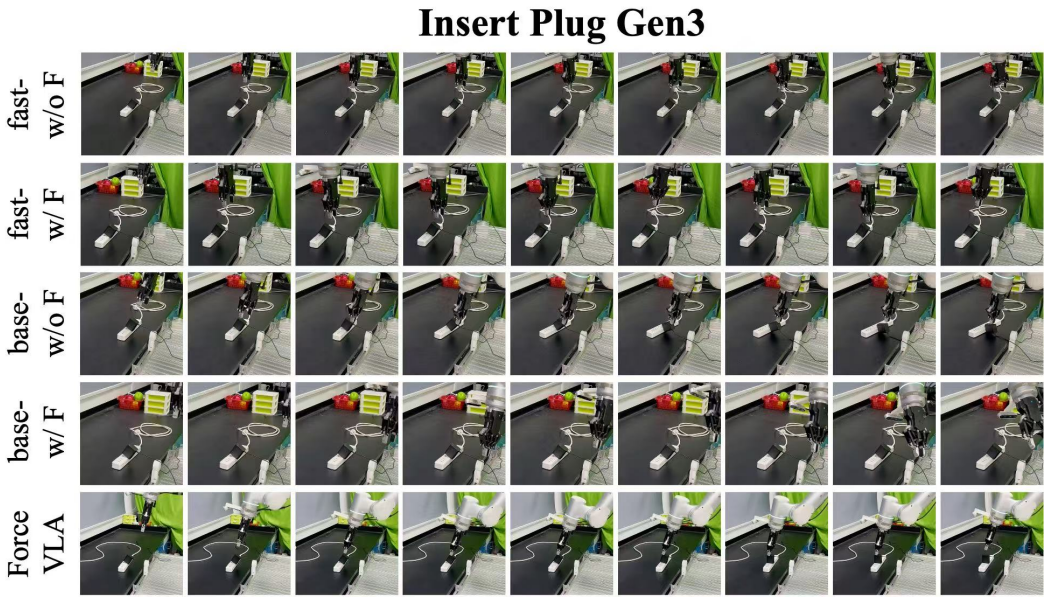


Figure 16: Key frames from Insert Plug Generalization task 3 videos.

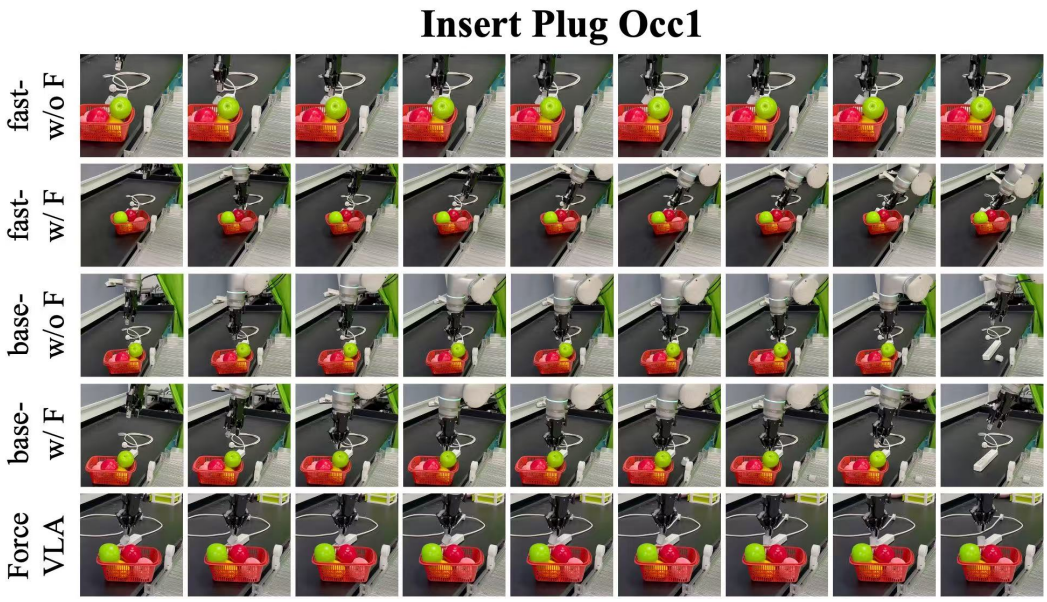


Figure 17: Key frames from Insert Plug Occlusion task 1 videos.

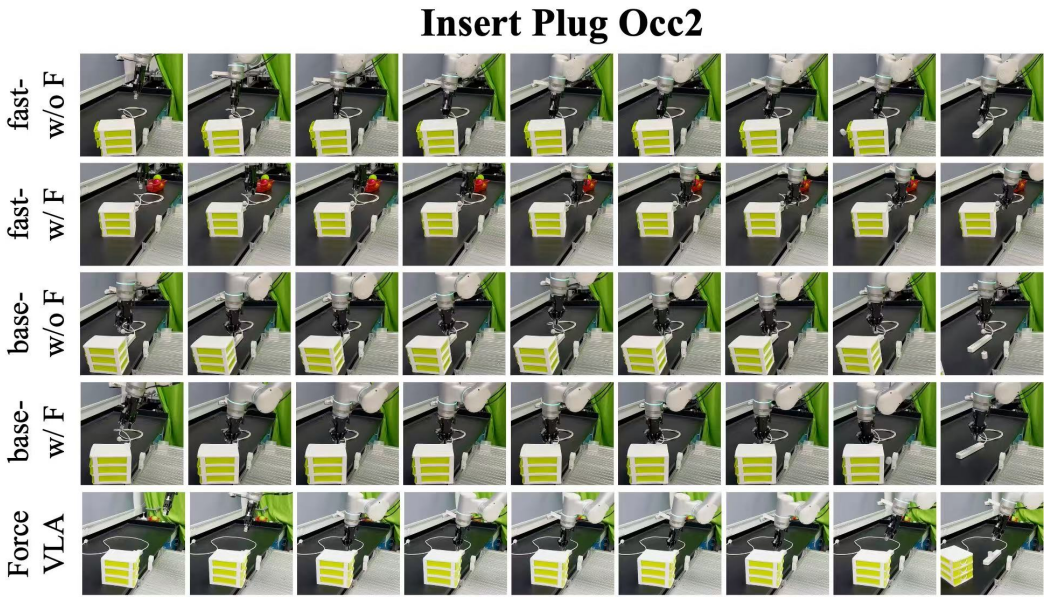


Figure 18: Key frames from Insert Plug Occlusion task 2 videos.

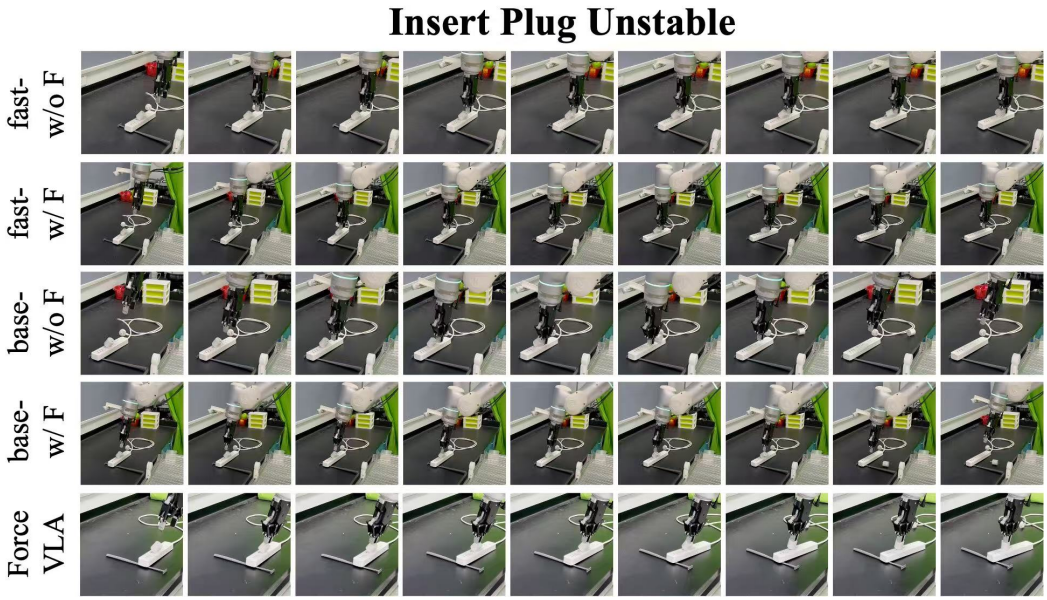


Figure 19: Key frames from Insert Plug Unstable task videos.