



Facet-0: A Robotic Foundation Model for Contact-Rich Precise Manipulation

Haoyuan Deng^{*1}, Haichao Liu^{*1}, Wenkai Guo^{*1}, Yuan Ling¹, Zaijia Yang¹, Yuanjiang Xue¹, Haosheng Sun¹, Liangzi Wang¹, and Ziwei Wang^{†1}

¹PINE Lab, Nanyang Technological University, Singapore

^{*}Equal contribution. [†]Corresponding author.

Abstract

Real-world robotic assembly at sub-millimeter tolerances demands spatial precision, compliant interaction, and robustness to contact failures. We present Facet-0, a robotic foundation model that predicts and values the contact consequences of its actions. Facet-0 unifies multimodal representation learning and reinforcement learning (RL) post-training around a joint action–wrench proposal: a causal wrench history is aligned with vision–language semantics and kinematic state, and flow matching generates each action chunk together with the future wrist-wrench profile it is expected to induce. Deployment rollouts train a distributional Action–Wrench Critic to distinguish motions with similar task progress but different contact outcomes, while phase-aware rewards and contact-selective credit concentrate policy improvement on decisive interactions. To accommodate part-specific dynamics, a lightweight bounded actor reuses the frozen representation for on-robot adaptation; RL remains defined over executable Cartesian actions, while an auxiliary wrench head preserves predictive, non-commanded action–contact coupling. Trained on ManuFacet-1K, a 1,000-hour force-synchronized corpus spanning three embodiments and multiple manufacturing cells, the bounded task-adapted system reaches 82% mean success on five sub-millimeter computer-assembly tasks, compared with 15% for the strongest baseline, with 0.5 mm placement accuracy and 50 ms command latency.

Project page: <https://pine-lab-ntu.github.io/facet-0/>

Correspondence: ziwei.wang@ntu.edu.sg

Keywords: robot foundation models, force-aware precision assembly, real-world reinforcement learning

1 Introduction

Contact-rich manufacturing manipulation requires simultaneous spatial precision, tactile regulation, throughput, and reliability. A robot must infer the geometry of a tight interface, regulate forces during insertion or seating, and recover from incipient jams without losing the task state. These demands expose a persistent limitation of general-purpose robot learning: broad visual competence does not by itself provide the fine-grained multimodal control required at sub-millimeter tolerances.

Existing approaches address only fragments of this problem. General-purpose vision-language-action (VLA) models [7, 8, 18, 37] provide broad task coverage, but their low-rate, position-oriented interfaces underrepresent the force and state history needed for contact regulation. Classical compliant controllers [21, 42, 46] can stabilize contact, yet are commonly hand-tuned and weakly coupled to learned multimodal representations. Force-aware VLA variants [19, 22, 49, 52] improve contact perception, but often leave the learning loop open: failures are detected at deployment without a principled mechanism for assigning credit to the corrective interaction that follows.

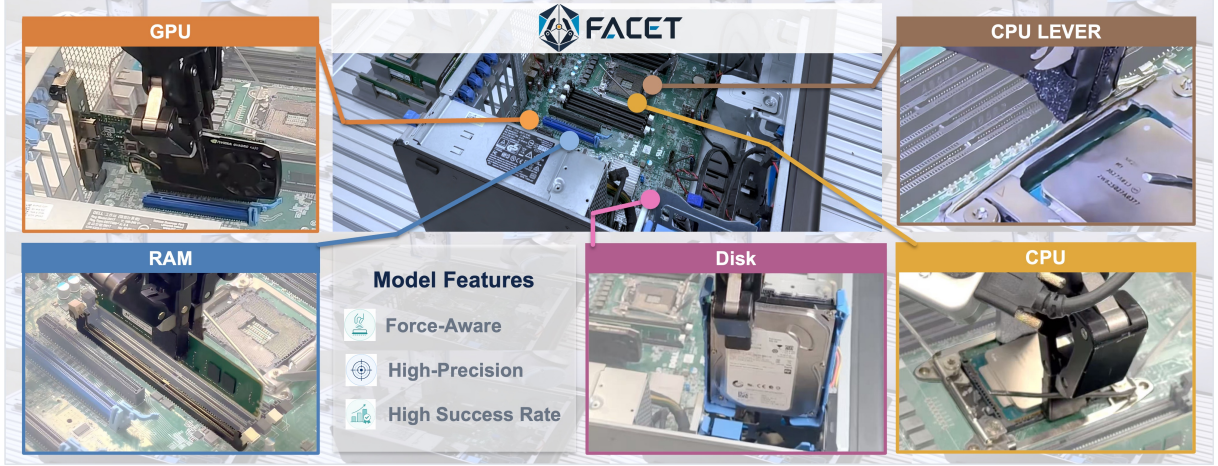


Figure 1. Qualitative overview of Facet-0 in precision computer assembly. The center shows the robotic workcell, and the surrounding panels show representative GPU, RAM, Disk, CPU, and CPU LEVER operations. Across these contact-rich tasks, Facet-0 targets precise alignment, force-aware execution, and reliable completion.

We present Facet-0 around a single learned object: a semantic–contact representation that connects the task the robot intends to perform with the interaction its motion is expected to produce. Its data foundation is ManuFacet-1K, a 1,000-hour corpus of force-synchronized demonstrations and closed-loop rollouts collected across three robot embodiments, two chassis families, and multiple manufacturing cells under a unified sensing and annotation protocol. Multi-view images and language provide task semantics, while kinematic state and a causal wrench window describe robot geometry and recent contact. Structured attention shares the semantic prefix between language supervision and interaction prediction without exposing either target path to the other. A flow-matching action expert then generates a joint proposal containing both the Cartesian action chunk and its anticipated wrench profile. Predicting wrench as part of the proposal makes contact consequence explicit in the learned representation, while only the action component is executed.

Deployment experience improves this same representation through value-guided RL post-training. A vision–language judge segments rollouts into sub-goals and supplies completion and contact-phase labels for a reward that balances progress, efficiency, and wrench-envelope violation. A distributional Action–Wrench Critic evaluates the joint proposal, allowing two motions with equal geometric progress to receive different values when one aligns cleanly and the other jams. Contact-selective credit ranks interactions within contact and free-space regimes and conditions the shared action expert on the valuable subset. When part dynamics change, the aligned representation is compressed into a frozen FACET token that conditions a bounded local actor together with measured robot state and the frozen policy reference. Its TD3 critics evaluate only the executable Cartesian action, while an auxiliary future-wrench head remains predictive and non-commanded. Thus representation learning, value assignment, and local transfer preserve one action–contact semantics rather than introducing independent policy components. On a five-task computer-assembly benchmark, the bounded task-adapted Facet-0 achieves 82% mean success (10% for $\pi_{0.5}$ and 4% for GR00T N1.7 [36]), 0.5 mm placement accuracy, and 50 ms command latency (150 ms for $\pi_{0.5}$). Controlled variants progress from 16% with semantic–contact alignment alone to 38% after value-guided refinement and 82% with local adaptation. Against matched advantage-weighted behavior cloning on the same deployment corpus, critic-guided post-training lowers intervention from 47% to 24% and raises recovery from 44% to 81%. Adaptation to an unseen memory module uses ten demonstrations and three hours of training on 6.6% of the parameters, reaching 45% versus 5% for the strongest baseline.

Our contributions are as follows.

- **Force-synchronized manufacturing data.** ManuFacet-1K provides 1,000 hours of demonstrations and closed-loop rollouts across three embodiments and multiple manufacturing cells.

- **A semantic–contact representation.** Structured attention and joint action–wrench flow matching connect task intent to anticipated contact consequences.
- **Value-guided RL post-training.** An Action–Wrench Critic and contact-selective credit refine deployment behavior, while a bounded local actor adapts the frozen representation to new parts.
- **Real-world assembly results.** The bounded task-adapted system reaches 82% mean success on five sub-millimeter tasks, with 0.5 mm accuracy, 50 ms latency, and improved recovery and unseen-part transfer.

2 Related Work

2.1 Vision-Language-Action Foundation Models

Modern VLA policies combine a pretrained vision–language backbone with an action decoder, spanning autoregressive tokens, diffusion, and flow matching over action chunks [7, 8, 10, 18, 26, 29, 37]. Knowledge-insulated co-training preserves semantic priors while the action expert learns from robot trajectories [14], and efficient action tokenization improves the practical control rate of this family [40]. These models generalize broadly across scenes and instructions, but their learned output is usually motion alone: contact, when available, is treated as an observation rather than as a consequence attached to the proposed action. That distinction matters in precision assembly, where two motions can reach the same pose while producing a clean insertion or a damaging jam. Facet-0 retains the semantic breadth of a generalist backbone but changes the object learned by the action expert. It fuses a causal wrench history with visual–language and kinematic context into a semantic–contact representation, then generates action and anticipated wrench jointly. The resulting proposal remains instruction-grounded while exposing interaction quality to the value-guided post-training objective of Sec. 5.

2.2 Force- and Tactile-Aware Policies

A rapidly growing body of work integrates contact signals into learned policies. ForceVLA routes six-axis wrench through a force-aware mixture of experts [49]. TA-VLA aggregates wrist force/torque history and uses auxiliary wrench prediction to improve the learned interaction representation [52]. Tactile-VLA couples tactile-aware reasoning with hybrid position–force control [22], while FoAR uses force feedback in a reactive contact policy [19]. FD-VLA distills a force representation for sensorless inference, and ForceFlow combines force-aware fusion with contact-driven flow matching and force prediction [51, 53]. These results establish the value of contact supervision, but force is usually introduced as an input, auxiliary target, or local control cue within imitation learning. Facet-0 instead attaches anticipated wrench to the proposed motion and lets deployment value score that joint proposal. Contact therefore influences which interactions the policy improves, not only what it observes.

2.3 Contact Control and Compliant Assembly

Classical contact control solved the actuation half of this problem decades ago. Hybrid force-position control partitions the task frame into force-regulated and position-regulated directions [34, 42], impedance and admittance control shape the manipulator’s apparent dynamics at contact [5, 21], variable impedance adapts those dynamics online [1], and energy-tank formulations make passivity, and therefore interaction safety, a structural property rather than a tuning outcome [15, 27]. Peg-in-hole and its relatives have been studied under this apparatus for as long as robots have assembled anything [23, 45, 46]. These controllers offer principled contact regulation under their design assumptions, but transferring them across workpieces often requires task-frame identification and parameter retuning. Facet-0 addresses the complementary learning problem: it anticipates and values contact consequences while leaving compliant control and explicit workspace and per-step motion limits at the execution layer.

Table 1. Comparison with manipulation corpora using the scale reported by each source. Native units are retained because trajectory counts cannot be converted reliably to hours without duration statistics. “Mixed” denotes heterogeneous constituent datasets and “n/r” denotes information not reported in the cited source.

Dataset	Reported scale	Sync. F/T	Temporal labels	Long-horizon
Open X-Embodiment [38]	1.4M trajectories	mixed	mixed	mixed
AgiBot World [2]	2,976.4 h	n/r	sub-step	✓
RoboCasa [35]	100K+ trajectories	n/r	n/r	✓
RoboMIND [48]	305.5 h	n/r	10K frame-level	n/r
DROID [25]	350 h	n/r	n/r	n/r
FMB [32]	≈31 h	✓	stage	✓
REASSEMBLE [44]	≈13 h	✓	action/skill	✓
ManuFacet-1K (ours)	≈1,000 h	✓	contact phase	✓

2.4 Learning from Deployment Experience

Real-world RL has matured into post-training recipes for generalist policies. SERL and HIL-SERL established sample-efficient off-policy learning with human corrections on real robots [31, 33], and a follow-on line attacks the labor those corrections cost, pruning transitions by their influence on policy entropy [12] or making the intervention itself agentic, triggered from value dynamics rather than by a human takeover [11]. RL-100 chains imitation, offline RL and brief online RL into deployment-grade reliability [28]. RECAP scales advantage-conditioned policy extraction to VLA post-training [41], whereas VLA-RL scales online RL for autoregressive VLAs [30]; a recent survey maps the space [13]. A parallel line adapts frozen policies without touching their weights, steering latent noise [47], learning residual corrections [3, 24, 43, 50], or stabilizing on-robot fine-tuning [9]. We adopt their shared principle, that a learned value estimate rather than the data distribution should select experience, and depart twice: our Action–Wrench Critic scores the predicted pair, so gentle alignment outranks ramming even when both succeed, and the policy is improved in action–wrench space, changing behavior *during* contact rather than only the aim before it. Intervention rate is correspondingly a metric we report, not only a training cost.

2.5 Manipulation Datasets

Open X-Embodiment aggregates over one million episodes across embodiments [38], DROID contributes 350 hours of diverse scenes [25], AgiBot World scales to thousands of hours on standardized hardware [2], and RoboCasa and RoboMIND add simulated and multi-embodiment coverage [35, 48]. These resources prioritize behavioral breadth rather than a corpus organized around synchronized six-axis wrench supervision and contact-phase annotation. Assembly-focused resources such as REASSEMBLE and FMB provide contact-rich trajectories or reproducible assembly benchmarks, but at substantially smaller scale [32, 44]. ManuFacet-1K complements the first group and scales the second: it is force-synchronized throughout, collected across three embodiments and two chassis families at multiple centers under one hardware and teleoperation specification, and annotated with a skill-phase taxonomy reused for reward decomposition and refinement conditioning.

3 The ManuFacet-1K Dataset

ManuFacet-1K is an approximately 1,000-hour, force-synchronised corpus for precision manipulation. It combines demonstrations and closed-loop rollouts across multiple sites and installation tasks, with a shared state representation and contact-phase vocabulary. This section specifies the corpus and summarises its collection and curation; implementation thresholds and quality-control details are deferred to Appendix B.

3.1 Corpus Composition

The corpus covers four installation tasks in the proportions shown in Fig. 2: CPU (37.3%), RAM (21.9%), Disk (23.4%), and GPU (17.3%), spanning diverse contact geometries and force regimes.

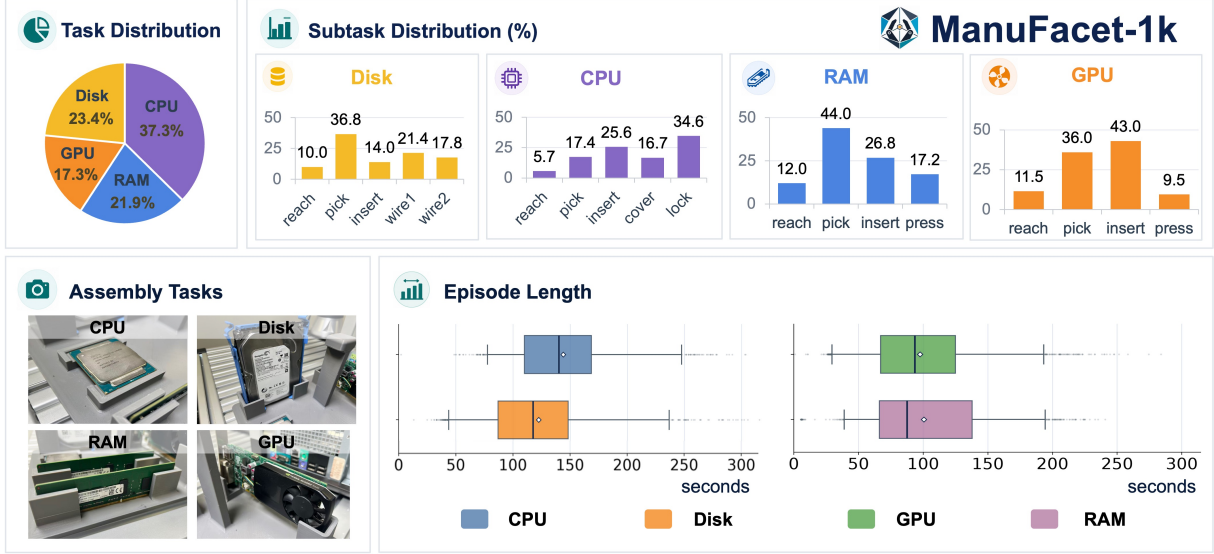


Figure 2. ManuFacet-1K at a glance. The ManuFacet-1K corpus contains approximately 1,000 hours of force-synchronized demonstrations and closed-loop rollouts, collected through a multi-center network across four precision installation tasks (CPU 37.3%, RAM 21.9%, Disk 23.4%, GPU 17.3%), at 0.5 mm action resolution.

Each cell uses a torque-controlled arm with parallel gripper, wrist six-axis force/torque sensing, three RGB cameras, and joint proprioception; kinesthetic teleoperation with force feedback supplies demonstrations in which force regulation is intentional. Every frame stores the 13-D state

$$s_t = [x_t \in \mathbb{R}^6; g_t \in \mathbb{R}; w_t \in \mathbb{R}^6], \quad (1)$$

combining end-effector pose, gripper opening and wrench.

The curated corpus comprises approximately 1,000 hours of open-loop teleoperated demonstrations and closed-loop policy rollouts with human intervention. Action resolution is 0.5 mm. VLM-generated sub-task/instruction pairs accompany each segment and are reused for pretraining, reward decomposition and adaptation. The release includes a data card, pseudonymised center identities, removed personally identifying content, and curation code.

3.2 Skill-Phase Taxonomy and Statistics

We define seven phases over manufacturing skills: *approach*, *align*, *insert*, *press*, *seat*, *fasten*, and *retreat*. The five contact-intended phases (*align*, *insert*, *press*, *seat*, *fasten*) are designated critical because precision determines the outcome. Contact-phase frames are a minority of the corpus but carry most outcome information; peak forces are bimodal, with a low-force alignment mode and a seating mode roughly an order of magnitude higher, depending on the part. Boundaries are marked during collection and verified offline by the VLM pass, with disagreements sent for re-annotation. This shared vocabulary makes phase-indexed frames addressable for pretraining, reward decomposition and action-wrench conditioning. It is distinct from the evaluation skill verbs in Sec. 6: phases describe temporal interaction regimes within a trajectory, whereas verbs name benchmark sub-goals, each of which may traverse multiple phases.

3.3 Collection and Curation

Episodes are streamed from a multi-center network under a shared sensing, teleoperation and state specification. Curation consists of five steps: (i) timestamp-based multimodal synchronization; (ii) segment-level validity extraction; (iii) saliency-based action-chunk sampling; (iv) VLM-assisted sub-task and instruction supervision; and (v) anomaly correction with corrected trajectories retained for the data flywheel. Full rates, thresholds, annotator protocol and reflow rules are given in Appendix B.

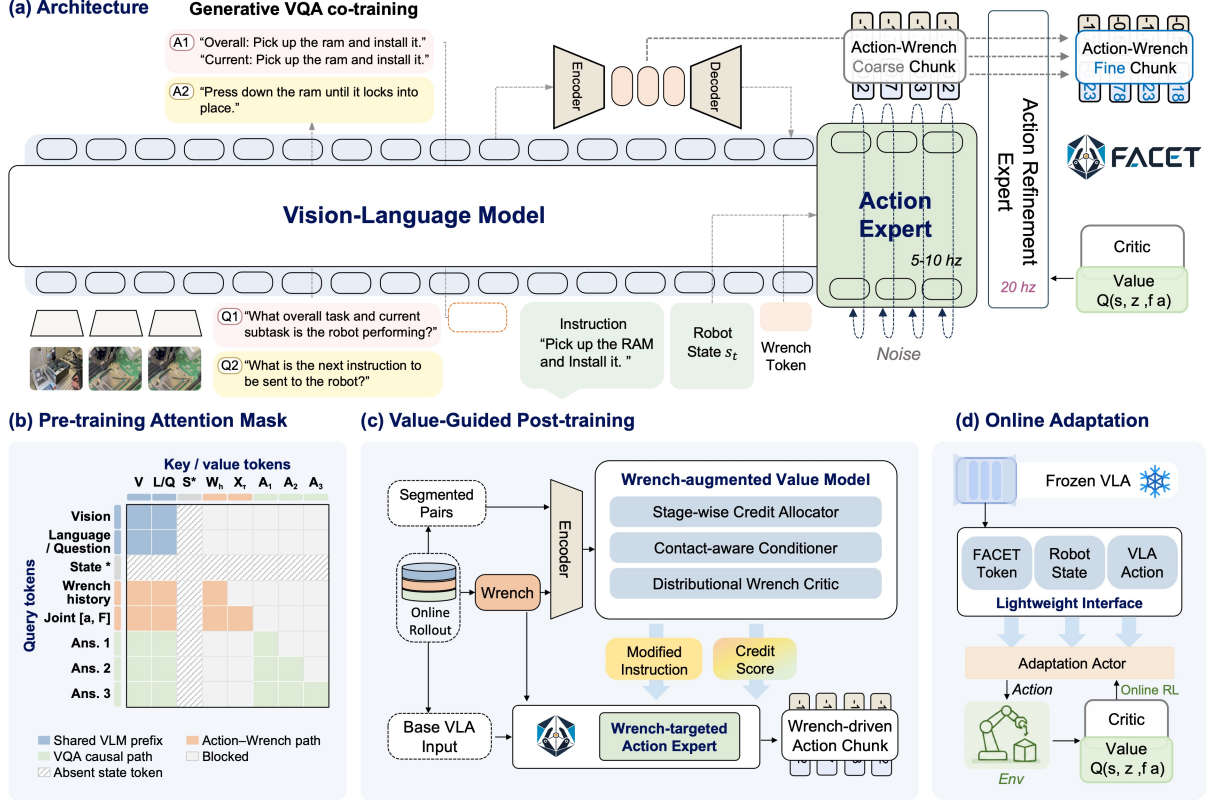


Figure 3. Overview of Facet-0. **(a)** Visual–language context, robot state, and wrench history form a shared semantic–contact representation for coarse-to-fine action–wrench decoding. **(b)** Structured attention shares the visual–language prefix while separating the action–wrench and causal VQA paths. **(c)** Phase-segmented rollouts train an Action–Wrench Critic; contact-selective credit then refines the shared action expert. **(d)** A bounded local actor reuses the frozen representation and policy reference; its critics evaluate executable actions, while its auxiliary wrench prediction remains non-commanded.

4 Semantic–Contact Alignment

Precision assembly couples task semantics with contact dynamics that are only partially visible near insertion. Since wrist force/torque reports interaction only after execution, we model each proposed motion jointly with its anticipated wrench, thereby aligning semantic intent with contact consequences.

4.1 Multimodal Grounding

The base policy pairs a PaliGemma vision–language backbone [6] with a flow-matching action expert [29]. At control step t , the observation contains three synchronized RGB views $I_t^{1:3}$, a language instruction ℓ , and the robot state of Eq. (1). We separate that state into the end-effector pose $x_t \in \mathbb{R}^6$, gripper opening $g_t \in \mathbb{R}$, and measured wrist wrench $w_t = [f_t; m_t] \in \mathbb{R}^6$, where f_t and m_t are the three-axis force and moment. The kinematic input is $s_t^{\text{kin}} = [x_t; g_t] \in \mathbb{R}^7$, while $W_{t-K+1:t} = [w_{t-K+1}, \dots, w_t] \in \mathbb{R}^{K \times 6}$ stores the latest K wrenches. We use $K = 10$ so that contact onset, persistent off-axis moment, and rising wrench under stalled motion can be represented as temporal events rather than isolated samples.

The semantic and contact streams are encoded and fused as

$$h_t^{\text{VL}} = E_{\text{VL}}(I_t^{1:3}, \ell),$$

$$h_t^c = F_{\text{fuse}}\left(h_t^{\text{VL}}, E_s(s_t^{\text{kin}}), E_w(W_{t-K+1:t})\right), \quad (2)$$

Here E_{VL} , E_s , and E_w are respectively the vision–language, kinematic-state, and wrench-history encoders; F_{fuse} is the learned cross-modal fusion map; and h_t^c is the semantic–contact representation. Conditioned on

h_t^c , the flow-matching policy π_θ with parameters θ samples a horizon- H joint proposal $Y_t \sim \pi_\theta(\cdot \mid h_t^c)$. We write this proposal as $Y_t = [A_{t:t+H-1}, \widehat{W}_{t+1:t+H}] \in \mathbb{R}^{H \times 13}$, where $A_{t:t+H-1} \in \mathbb{R}^{H \times 7}$ contains Cartesian-pose and gripper commands and $\widehat{W}_{t+1:t+H} \in \mathbb{R}^{H \times 6}$ contains the wrist wrenches predicted after those commands, expressed in physical units after reversing the training normalization. Thus row k pairs a_{t+k} with its anticipated consequence $\widehat{w}_{t+k+1}$. We use $H = 50$. Only A is executable; $\widehat{W}$ is an anticipated contact consequence, not a force command.

As shown in Fig. 3(a), the action expert produces a coarse action–wrench chunk at 5–10 Hz and the refinement expert updates it at 20 Hz without changing the joint parameterization. The structured mask in Fig. 3(b) aligns the two learning paths without target leakage: visual and language/question tokens form their shared prefix; the action–wrench path additionally reads $W_h = E_w(W_{t-K+1:t})$ and the noised joint chunk X_τ defined below, whereas the VQA path attends causally to preceding answer tokens. Cross-attention between their targets is blocked, and robot state is injected only into the action expert, as indicated by the hatched state row and column.

4.2 Predictive Interaction Learning

Each demonstration provides a clean data target $Y_t^{\text{data}} = [A_{t:t+H-1}^{\text{data}}, W_{t+1:t+H}^{\text{data}}] \in \mathbb{R}^{H \times 13}$, whose columns contain executed actions and the measured wrenches that follow them. Conditional flow matching [29] interpolates this target with Gaussian noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ of the same shape, where $\mathbf{I}$ is the identity covariance. For a sampled interpolation time $\tau \in [0, 1]$, let $X_\tau = (1 - \tau)Y_t^{\text{data}} + \tau\epsilon$ and $U_\tau = \epsilon - Y_t^{\text{data}}$. The learned velocity field v_θ is optimized by

$$\begin{aligned} \mathcal{L}_{\text{AW}}(\theta) = \mathbb{E} \left[\left\| v_\theta^a(X_\tau, \tau \mid h_t^c) - U_\tau^a \right\|_2^2 \right] \\ + \lambda_{\text{pre}} \mathbb{E} \left[\left\| v_\theta^w(X_\tau, \tau \mid h_t^c) - U_\tau^w \right\|_2^2 \right]. \end{aligned} \quad (3)$$

The superscripts a and w select the seven action and six wrench columns, respectively; validity normalization is applied independently to the available entries in each slice. The scalar λ_{pre} balances wrench prediction against action learning, and the expectation is over demonstrations, τ , and ϵ . At inference, integrating v_θ backward from noise at $\tau = 1$ to data at $\tau = 0$ produces the joint proposal Y_t defined above; consequently, lowering the wrench error requires the policy to encode how its proposed motion changes contact.

The semantic objective $\mathcal{L}_{\text{VQA}}$ is answer-token cross-entropy for two questions: the overall/current sub-task and the next instruction. Separate optimizers alternate between $\mathcal{L}_{\text{AW}}$ and $\mathcal{L}_{\text{VQA}}$ rather than mixing their heterogeneous scales. Combined with the structured mask, this trains a shared semantic prefix while preserving distinct causal paths for interaction and language supervision.

5 Value-Guided Policy Refinement

The aligned policy proposes plausible interactions but cannot rank them by task utility. Deployment rollouts supply that ranking through sub-goal completion, contact quality, and recovery from failure. Within the aligned policy, the coarse expert supplies a horizon-scale joint proposal and the refinement expert updates that same action–wrench object near contact. We learn a value function over these proposals and use it to refine the shared generative policy. For part-specific transfer, a downstream local actor instead consumes the frozen representation and policy reference; its critic acts only on the seven-dimensional executable action and does not alter the global proposal space.

5.1 Interaction Valuation

A vision–language judge Ω segments each rollout into sub-goals, marking completion with $\Omega_t \in \{0, 1\}$ and labeling the contact phase φ_t . Let $\|w_t\|_{\varphi_t}$ denote the six-axis wrench normalized per axis by the envelope configured for phase φ_t , so any value above one is a violation, and let Δt be the time spent in the

current sub-goal against its budget $T_{\max}$. Progress and contact then share a single per-step signal,

$$r_t = \Omega_t \left(1 + \beta \left[1 - \frac{\Delta t}{T_{\max}} \right]_+ \right) - \alpha [\|w_t\|_{\varphi_t} - 1]_+, \quad (4)$$

where $[x]_+ = \max(x, 0)$. Thus completing a sub-goal earns credit, completing it sooner earns more, and forcing through it earns less; $\alpha, \beta \geq 0$ set the exchange rate.

Using the notation of Sec. 4, $Y_t \in \mathbb{R}^{H \times 13}$ is the joint action–wrench proposal and h_t^c is the semantic–contact representation. The critic $Z_\psi(h_t^c, Y_t)$ predicts the return distribution of executing the action component of Y_t [4]; its mean $Q_\psi = \mathbb{E}[Z_\psi]$ is the Action–Wrench Critic, and $V_\psi(h_t^c) = \mathbb{E}_{Y \sim \pi_\theta}[Q_\psi(h_t^c, Y)]$ is the value of the policy’s own proposal. Training is temporal difference on stored rollouts,

$$\mathcal{L}_Q(\psi) = \mathbb{E} \left[\mathcal{D}_{\text{dist}}(Z_\psi(h_t^c, Y_t), r_t + \gamma(1 - d_t)Z_{\bar{\psi}}(h_{t+1}^c, Y_{t+1})) \right], \quad (5)$$

where $\mathcal{D}_{\text{dist}}$ is the distributional regression loss, γ is the discount, $d_t \in \{0, 1\}$ marks a terminal transition, $\bar{\psi}$ denotes target parameters, and $Y_{t+1} \sim \pi_\theta(\cdot | h_{t+1}^c)$ is the successor proposal. The factor $1 - d_t$ removes the bootstrap at terminal transitions.

Two properties of this critic matter downstream. Reading the predicted wrench alongside the action lets it separate a clean insertion from a jam stalled at the same depth, which are indistinguishable by geometric progress alone. Predicting a distribution rather than a mean preserves the success and failure modes that contact-rich outcomes split into, instead of averaging them into a value no frame actually attains. Rollouts are stored as adjacent, phase-segmented transitions and replayed offline, shown as “Segmented Pairs” in Fig. 3(c).

5.2 Contact-Selective Credit

Ranking frames by return favors those that make visible progress, and decisive contact is not among them: it occupies few frames and advances measured progress slowly. We therefore score each frame by a short-horizon, undiscounted credit

$$\delta_t^{(N)} = \sum_{j=0}^{N-1} r_{t+j} + V_\psi(h_{t+N}^c) - V_\psi(h_t^c), \quad N < H, \quad (6)$$

which measures how far a frame outruns the progress rate the critic already expects at that point. Here N is the credit horizon, chosen shorter than the action-chunk horizon H . Being undiscounted, $\delta_t^{(N)}$ is a within-rollout credit score rather than a Bellman-consistent N -step advantage, which is all the label below requires.

A single global threshold on $\delta_t^{(N)}$ would still select mostly free-space frames. We instead split frames by contact regime $\rho_t \in \{\text{contact}, \text{free}\}$ and take the top-ranked fraction within each. If q_{ρ_t} is the within-regime score threshold, the positive-interaction label is $b_t = \mathbf{1}[\delta_t^{(N)} \geq q_{\rho_t}]$, where $\mathbf{1}[\cdot]$ is the indicator function. The normalized contact intensity $c_t \in [0, 1]$ then weights the refinement loss,

$$\mathcal{L}_{\text{post}}(\theta) = \mathbb{E}_t \left[(1 + \lambda c_t) \mathcal{L}_{\text{FM}}(\theta; Y_t^{\text{data}}, h_t^c, b_t) \right], \quad (7)$$

where $\mathcal{L}_{\text{FM}}$ is the per-sample action–wrench flow-matching loss of Eq. (3), $\lambda \geq 0$ sets the contact upweighting, and Y_t^{data} is the executed-action/realized-wrench target defined in Sec. 4.2. In contrast, Y_t without a superscript always denotes the policy’s predicted joint proposal scored by the critic.

The label reaches the policy as a short tag appended to the instruction: positive frames train the guided branch, and the unconditional branch trains on all frames with the tag dropped. This is the contact-selective credit mechanism of Fig. 3(c), where $\delta_t^{(N)}$ is the displayed credit score, b_t is delivered as a tagged instruction to the shared action expert, and Q_ψ supplies the value that guides the 20 Hz refinement expert.

5.3 Local Policy Adaptation

Part changes primarily alter local contact dynamics while leaving task semantics and coarse geometry intact. The frozen bottleneck encoder E_z compresses the semantic–contact representation h_t^c learned in Sec. 4 into the FACET token $z_t = E_z(h_t^c)$. The frozen policy also supplies a reference action $\tilde{a}_t \in \mathbb{R}^7$. Learned projections followed by layer normalization map z_t , the kinematic state $[x_t; g_t]$, the measured wrench w_t , and $\tilde{a}_t$ to modality embeddings e_t^z , e_t^p , e_t^w , and e_t^{ref} , respectively. Their concatenation $e_t = [e_t^z; e_t^p; e_t^w; e_t^{\text{ref}}]$ is the lightweight actor interface in Fig. 3(d), with dimensions detailed in Appendix D.3.

Let f_η be the shared actor with parameters η . Its two heads return a raw action activation $u_t^a \in \mathbb{R}^7$ and the next-wrench prediction $\hat{w}_{t+1} \in \mathbb{R}^6$, written $f_\eta(e_t) = [u_t^a, \hat{w}_{t+1}]$. A componentwise tanh followed by affine rescaling maps u_t^a into the task-specific box $[a_{\min}, a_{\max}]$, where $a_{\min}, a_{\max} \in \mathbb{R}^7$, to obtain the bounded absolute Cartesian target $a_t \in \mathbb{R}^7$. The existing safety interface applies workspace and per-step motion limits immediately before execution. The actor is trained with

$$\begin{aligned} \mathcal{L}_{\text{actor}}(\eta) = & -\mathbb{E}[Q_{\xi_i}(e_t, a_t)] + \lambda_{\text{BC}} \mathcal{L}_a(a_t, a_t^{\text{human}}) \\ & + \lambda_{\text{pred}} \mathcal{L}_w(\hat{w}_{t+1}, w_{t+1}). \end{aligned} \quad (8)$$

The expectation is over online actor states. The first term improves the actor under one of the twin TD3 critics [17], denoted Q_{ξ_i} for $i \in \{1, 2\}$ with parameters ξ_i ; both critics have domain (e_t, a_t) and evaluate only the executable seven-dimensional action. Actor updates query the bounded proposal, whereas critic regression uses the safety-filtered command stored with each transition. In the remaining terms, a_t^{human} is a successful intervention action, $\mathcal{L}_a$ is its action-anchor loss, and $\mathcal{L}_w$ compares the predicted wrench with the measured next-step target w_{t+1} ; λ_{BC} and λ_{pred} are their respective weights.

The frozen reference $\tilde{a}_t$ is a conditioning prior, not a residual base. The predicted wrench is fully implemented and supervised as an auxiliary contact-consequence target, but is neither a TD3 action dimension nor a force command. Accordingly, the local critic is shown in Fig. 3(d) as $Q(e, a)$: it evaluates the actor interface and executable action, not the auxiliary prediction. At deployment only a_t is executed. The wrench head remains active but non-commanded, and compliant control closes its force loop using measured wrist F/T feedback.

6 Experiments

We evaluate Facet-0 as a deployed system rather than as a policy in isolation. The experiments answer four questions. (Q1) Does the full model beat generalist and force-aware VLAs on precision assembly, and how do semantic–contact alignment, value-guided refinement, and local adaptation contribute (Sec. 6.2)? (Q2) What does post-training from deployment experience buy that supervised post-training does not (Sec. 6.3)? (Q3) How does the local actor compare against other adaptation mechanisms, and how cheaply does it transfer to an unseen part (Sec. 6.4)? (Q4) Which uses of the wrench signal carry the gains (Sec. 6.5)?

6.1 Experimental Setup

Suite and protocol. Five host-computer installation tasks share one chassis fixture and decompose into 23 sub-goals over a seven-verb vocabulary (Fig. 4): nine free-space, and fourteen contact-critical at 0.10–0.30 mm clearance under per-phase safety limits $F_{\max}$. Chassis pose is randomized per trial, parts arrive by automated feeder, and every cell in the main tables is 20 trials under a pre-declared protocol with no re-runs.

Metrics and baselines. We report success, violation, intervention and recovery rates, peak off-axis force, cycle time and inference latency. The baselines are $\pi_{0.5}$ [7], $\pi_{0.5}+\text{F}$ (wrench as an extra input token), $\pi_{0.5}+\text{RECAP}$ -style (our implementation of the RECAP advantage-conditioning objective [41] on the $\pi_{0.5}$ backbone), GR00T N1.7 [36] and TA-VLA [52], all trained on ManuFacet-1K with matched budgets under one aligned protocol. For controlled Facet-0 variants, *Align* learns the semantic–contact representation, *+RL* adds value-guided post-training, and *Full* additionally enables local adaptation. Appendix A.1 gives

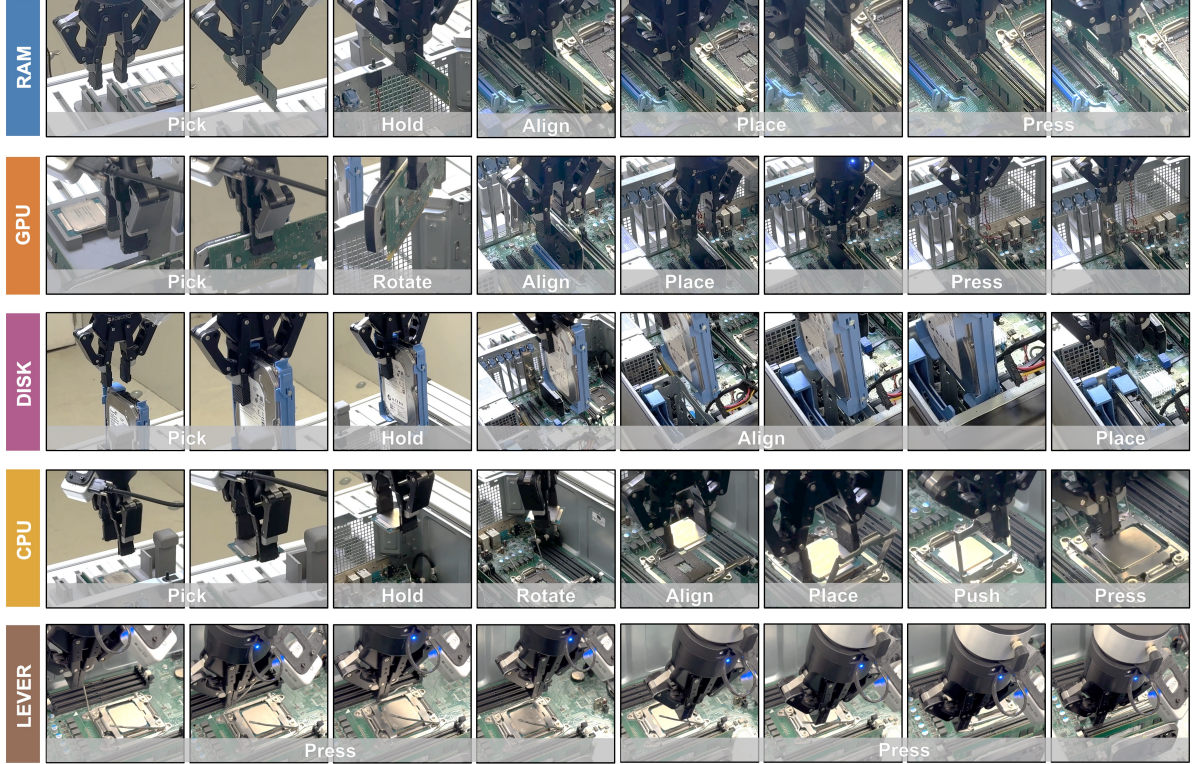


Figure 4. The five tasks of the assembly suite, each shown as the sequence of sub-goals the policy executes. Verbs are shared across tasks: *pick*, *hold* and *rotate* are free-space, while *align*, *place*, *push* and *press* are contact-critical and hand control to the refinement expert. The LEVER task closes the CPU socket lever in two sequential presses and is the only task with no free-space sub-goal. Clearances and force limits are in Appendix A, Table 5.

task chains, clearances, metric definitions and baseline configurations.

6.2 Main Comparison and Component Gains (Q1)

Table 2 reports task-level success for all eight methods, Fig. 5 resolves four align-containing tasks over the reported diagnostic sub-goals, and Table 7 in Appendix E gives the corresponding 17-row subset (13 contact-critical) of the 23-sub-goal suite. Averaged over the five tasks, Facet-0 attains 82% success against 15% for the strongest baseline, $\pi_{0.5}$ +RECAP-style, a $5.5\times$ improvement. Per-task margins over the best baseline range from +45 points on the LEVER task, the only task without a free-space sub-goal, to +75 points on GPU, with the remaining three tasks between +60 and +65.

Failures concentrate at contact. No baseline exceeds 35% on any task, and the diagnostic rows of Table 7 locate the measured deficit: the displayed *pick* rows run from 60% to 100% for every method, while *align* and *place* are where performance falls. Fig. 5 shows the same shape task by task, with every method near the rim on *pick* and the polygons moving inward on the sub-goals marked †. RAM and GPU *align* are both 10% for $\pi_{0.5}$ against 95% and 85% for Facet-0, localizing the largest gap to contact. Because a task succeeds only if every sub-goal in its chain succeeds, the task rows sit at or below the weakest link; the LEVER task, two contact-critical presses with no free-space step, is the hardest task in the suite and the only one where Facet-0 stays at 50%.

Feeling contact is not acting on it. $\pi_{0.5}$ +F and TA-VLA are the controls that matter for our central claim: both receive contact feedback, yet neither uses deployment value to distinguish the consequences of proposed actions. Both land in the 9–16% band shared by $\pi_{0.5}$ and the *Align* variant, five to nine times below the full system.

Representation, value, and adaptation are complementary. The controlled variants trace 16% $\rightarrow$ 38% $\rightarrow$ 82% without changing the evaluation protocol. *Align* establishes the joint action–wrench representation but has not yet used deployment outcomes to rank interactions. Adding value-guided RL contributes 22

Table 2. Task-level success rate (%) on the assembly suite, 20 trials per cell. **Left:** all eight methods, with baseline references as given in Sec. 6.1; best results are bold and second-best results are underlined, with ties retained. The mean covers all five tasks. **Right:** representative alignment views for the four insertion tasks.

Task	Baselines					Facet-0 (ours)		
	$\pi_{0.5}$	+F	+RECAP	GR00T	TA-VLA	Align	+RL	Full
RAM	10	15	35	10	10	15	<u>45</u>	95
CPU	5	5	15	5	20	20	<u>45</u>	85
Disk	25	20	20	5	30	30	<u>65</u>	95
GPU	10	5	5	0	5	10	<u>35</u>	85
LEVER	0	0	0	0	<u>5</u>	<u>5</u>	0	50
Mean (5 tasks)	10	9	15	4	14	16	<u>38</u>	82

Figure 5. Sub-goal success on four insertion tasks. Performance remains high in free-space *pick* and contracts at contact-critical sub-goals ($\dagger$); the two-step LEVER task is reported in Tables 2 and 7.

mean points; its gains concentrate on the four insertion tasks, improving them by 25–35 points, while the contact-only LEVER task remains unresolved. Local adaptation supplies the remaining part-specific correction: relative to +RL, the full model gains 50 points on RAM, 40 on CPU, 30 on Disk, 50 on GPU, and 50 on LEVER. Because task success requires the entire sub-goal chain, these gains measure completed assemblies rather than isolated contact events.

The improvement is consistent across task geometry. The full model reaches 85–95% on the four insertion tasks despite different parts, approach motions, and 0.10–0.30 mm clearances. LEVER remains the hardest at 50% because both commands are contact-critical presses and either failure invalidates the trial. This pattern separates breadth from precision: generalist baselines handle the shared free-space vocabulary, while the aligned representation, deployment value, and local contact correction determine whether that motion completes an assembly. Table 8 in Appendix F adds the deployment properties: a command every 50 ms against 150 ms for $\pi_{0.5}$, enabling the 20 Hz refinement expert, and a task completed 40% faster than an expert teleoperator.

Chain completion amplifies the contact bottleneck. The full-system task rate matches the weakest displayed contact-critical row in every chain: 95% for RAM and Disk, 85% for CPU and GPU, and 50% for LEVER (Table 7). By contrast, every displayed free-space *pick* row is 100%. Across the 13 reported contact sub-goals, the unweighted mean is 87%, five points above the 82% end-to-end mean. This gap follows from the chain criterion: every transition must succeed. It also explains why a bounded local correction at one limiting contact state can produce a much larger gain in completed assemblies.

Takeaway. Generalist VLAs retain strong performance on the displayed free-space *pick* sub-goals, while the measured deficit concentrates at contact. Supplying the wrench as an input does not close it, since $\pi_{0.5}$ +F and TA-VLA sit with $\pi_{0.5}$ and *Align*. The semantic–contact representation, value-guided refinement, and local adaptation jointly address that deficit.

6.3 Post-Training from Deployment Experience (Q2)

Against matched advantage-weighted behavior cloning (AWR [39]) on the Disk deployment corpus, our method changes success/intervention/recovery from 20/47/44% to 65/24/81% (Fig. 6).

Unlike AWR, which reweights the deployment corpus by a scalar advantage, our contact-selective credit ranks samples within interaction regimes. It therefore retains high-credit contact-recovery frames that are scarce in demonstrations (Sec. 5; Appendix F.1).

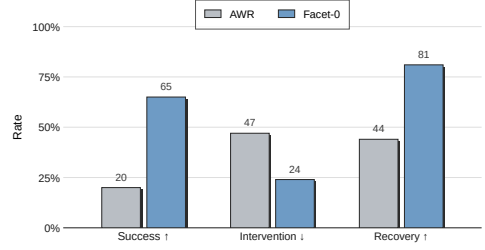


Figure 6. Disk post-training (Table 9).

Takeaway. Critic-guided post-training acts on the tail of the deployment distribution rather than on the average trajectory. Contact failures that vision cannot resolve remain separable in the wrench, so recovery frames retain positive credit instead of being filtered out (Appendix F.1).

6.4 Adaptation: Paradigms and Few-Shot Transfer (Q3)

Adaptation is evaluated along two axes. The first asks whether the adaptation actor is the right *mechanism* for on-robot learning, holding the task fixed. The second asks how cheaply the adapted system reaches an *unseen part*.

Adaptation dynamics. We first ask how the adaptation actor behaves as on-robot experience accumulates, against latent-noise steering of the frozen policy (DSRL [47]) under identical VLM reward. Fig. 7 tracks the two quantities that decide whether on-robot learning is affordable, over adaptation episodes. *Ours* (TD3+BC) is the adaptation actor of Sec. 5.3; *Ours* (HG-Dagger BC) replaces its actor-critic objective with interactive imitation of the same interventions, holding the interface and the bounded parameterization fixed. DSRL stays at the floor on both tasks for the episode budget shown. Both Facet-0 variants reach the ceiling, and the actor-critic variant ends at full success on either task, but they arrive differently: the imitation variant starts high and is comparatively flat, whereas the actor-critic variant starts lower, passes through a pronounced dip on RAM insert, and recovers. Autonomy rises in the trailing mean for both, which is the property that makes the procedure affordable, since the supervisor is consulted less as the episode count grows.

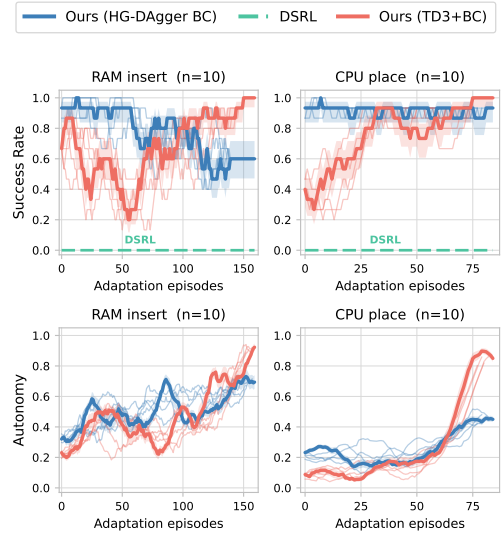


Figure 7. On-robot adaptation dynamics, $n = 10$ runs per curve. **Top:** success rate. **Bottom:** autonomy, the fraction of steps run without a supervisor takeover. Pale traces show representative individual runs; bold traces show a 15-episode trailing mean with a local 1σ band.

Adaptation paradigms. Fig. 8 summarizes this comparison at convergence and adds position-only residual adaptation [3]; raw measurements are in Appendix F, Table 10. Position-only residuals correct alignment, whereas our bounded actor adds an auxiliary wrench-consequence objective that preserves action-contact coupling. This prediction is not commanded; the right panel shows the resulting behavior. **Few-shot transfer to an unseen part.** Table 3 reports transfer to a memory module the system has never seen, with a different mass, stiffness, and latch signature; all methods receive ten demonstrations and three hours of training. Facet-0 reaches 9× the strongest baseline while updating a fifteenth of the parameters, and the two facts are connected. The baselines must relearn the task from ten demonstrations because they

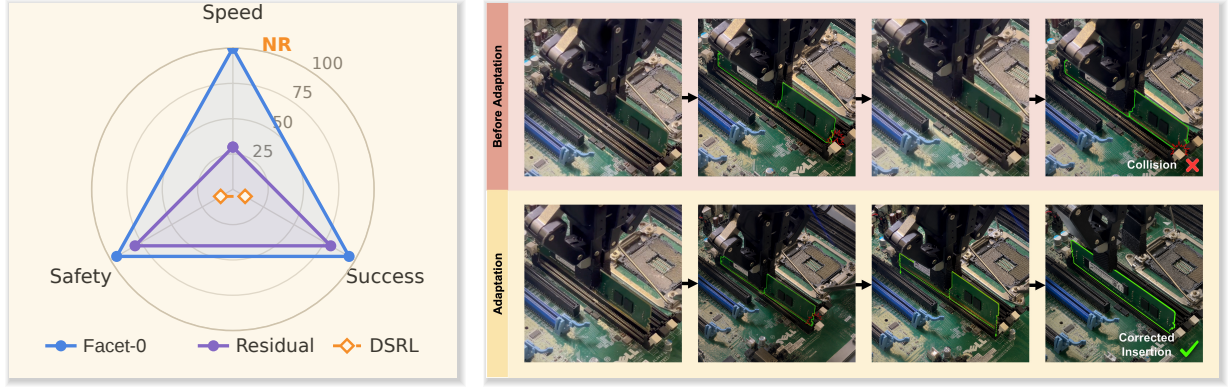


Figure 8. Adaptation on RAM insertion. **Left:** normalized comparison with position-only residual adaptation and DSRL; speed is time to 90% success, and NR means not reached. **Right:** after an off-axis collision, the adapted policy backs off, re-aligns and seats the module. Other axes and raw values are in Table 10.

have no interface that separates what changed, the contact dynamics of a heavier module with a stiffer latch, from what did not, the slot geometry, the approach, and the semantics of the instruction. Facet-0 trains only the adaptation actor, so the ten demonstrations are spent entirely on the part of the policy the unseen module actually invalidates.

Takeaway. Bounded action adaptation with auxiliary wrench prediction converts contact from a terminal failure into an actionable correction: after an off-axis collision the adapted policy backs off, re-aligns and seats the module on a second attempt (Fig. 8). The executable correction comes from the action head; the wrench head remains predictive and non-commanded, and the refinement expert stays frozen.

Table 3. Few-shot transfer to an unseen RAM module with a different mass, stiffness and latch signature. All methods receive 10 demonstrations and three hours of training. *Params.* is the fraction of policy parameters updated.

Method	Success (%) $\uparrow$	Params. $\downarrow$
$\pi_{0.5}$	5	100%
$\pi_{0.5}+F$	5	100%
GR00T N1.7	0	100%
TA-VLA	5	100%
Facet-0	45	6.6%

Table 4. Contact-role ablation on RAM insertion: the wrench signal is removed from prediction, interaction valuation, or local adaptation. The full model reproduces the RAM row of Table 2. Peak off-axis force is normalized to the force-free variant, so 1.0 $\times$ is the reference and lower is better.

Variant	Success (%) $\uparrow$	Peak force $\downarrow$
Full recipe	95	0.2$\times$
w/o predictive wrench	90	0.2 $\times$
w/o critic wrench	85	0.5 $\times$
w/o adaptation wrench	85	0.4 $\times$
w/o all contact roles	45	1.0 $\times$

6.5 Ablation Studies (Q4)

Contact-role ablation. Table 4 removes one action–wrench role at a time on RAM insertion. Joint removal is decisive: removing prediction, valuation, and adaptation cuts success by 50 points and returns peak off-axis force from 0.2 $\times$ to the force-free 1.0 $\times$. Individual removals differ by only one to two successes in 20 trials, so their ordering is not resolved, but the contact-quality pattern is consistent: predictive-wrench removal preserves the 0.2 $\times$ peak, whereas removing critic or adaptation wrench raises it to 0.5 $\times$ and 0.4 $\times$. Sec. 7 states the scope of this comparison.

Takeaway. Removing every predictive and evaluative role of wrench is the one degradation that is large relative to the measurement resolution. Individual removals trend consistently with complementary functions, but at 20 trials per cell they span only one to two trials and do not establish a ranking among prediction, valuation, and adaptation.

7 Discussion and Limitations

The training corpus spans three embodiments and two chassis families, while the controlled five-task evaluation uses wrist-sensed, parallel-gripper electronics assembly on a shared chassis fixture. Within this setting, the results support semantic–contact representation learning, value-guided refinement, and bounded adaptation, and the contact-role ablation distinguishes the complementary roles of prediction, valuation, and adaptation. Future work will broaden controlled evaluation to further end effectors, parts, and contact geometries, and explore force-conditioned policy distillation for cells without wrist-mounted six-axis sensing [53].

8 Conclusion

This work presents Facet-0, a robotic foundation model that links task semantics with anticipated contact and uses value-guided RL post-training to refine the behavior expressed through that representation. Its force-synchronized data, joint action–wrench prediction, Action–Wrench Critic, and bounded local adaptation connect perception, credit assignment, and execution at contact. On the five-task electronics-assembly suite, the full model reaches 82% mean success (15% for the strongest baseline), with 0.5 mm placement accuracy and 50 ms command latency. Controlled variants obtain 16% with semantic–contact alignment, 38% after value-guided refinement, and 82% with local adaptation. Relative to the matched AWR baseline, value-guided post-training reduces intervention from 47% to 24% and increases recovery from 44% to 81%.

The results indicate that one persistent semantic–contact representation can support both global policy improvement and local transfer: the Action–Wrench Critic values the policy’s joint proposal for global refinement, while a downstream bounded actor reuses the frozen representation and policy reference for an unseen module with ten demonstrations while updating 6.6% of the parameters. These findings are established on wrist-sensed electronics assembly, and extending them to further embodiments and contact regimes is ongoing work. We release the model, dataset, and training recipe to support that evaluation.

References

- [1] Fares J. Abu-Dakka and Matteo Saveriano. Variable impedance control and learning: A review. *Frontiers in Robotics and AI*, 7:590681, 2020.
- [2] AgiBot-World-Contributors, Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, et al. Agibot world colosseum: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025.
- [3] Lars Ankile, Zhenyu Jiang, Rocky Duan, Guanya Shi, Pieter Abbeel, and Anusha Nagabandi. Residual off-policy rl for finetuning behavior cloning policies. *arXiv preprint arXiv:2509.19301*, 2025.
- [4] Marc G. Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2017.
- [5] Cristian C. Beltran-Hernandez, Damien Petit, Ixchel G. Ramirez-Alpizar, Takayuki Nishi, Shinichi Kikuchi, Takamitsu Matsubara, and Kensuke Harada. Learning force control for contact-rich manipulation tasks with rigid position-controlled robots. *IEEE Robotics and Automation Letters*, 5(4):5709–5716, 2020.
- [6] 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.
- [7] Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. In *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pages 17–40. PMLR, 2025.

- [8] 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. In *Proceedings of Robotics: Science and Systems*, 2025.
- [9] Yuhui Chen, Shuai Tian, Shugao Liu, Yingting Zhou, Haoran Li, and Dongbin Zhao. ConRFT: A reinforced fine-tuning method for VLA models via consistency policy. In *Proceedings of Robotics: Science and Systems*, 2025.
- [10] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems (RSS)*, 2023.
- [11] Haoyuan Deng, Yitong Gao, Yudong Lin, Haichao Liu, Zhenyu Wu, and Ziwei Wang. UniIntervene: Agentic intervention for efficient real-world reinforcement learning. *arXiv preprint arXiv:2606.12372*, 2026.
- [12] Haoyuan Deng, Yudong Lin, Yuanjiang Xue, Haoyang Du, Qianzhun Wang, Boyang Zhou, Zhenyu Wu, and Ziwei Wang. E2HiL: Entropy-guided sample selection for efficient real-world human-in-the-loop reinforcement learning. *IEEE Robotics and Automation Letters*, 11(7):8084–8091, 2026.
- [13] Haoyuan Deng, Zhenyu Wu, Haichao Liu, Wenkai Guo, Yuquan Xue, Ziyu Shan, Chuanrui Zhang, Bofang Jia, Yuan Ling, Guanxing Lu, et al. A survey on reinforcement learning of vision-language-action models for robotic manipulation. *TechRxiv*, 2025.
- [14] Danny Driess, Jost Tobias Springenberg, Brian Ichter, Lili Yu, Adrian Li-Bell, Karl Pertsch, Allen Z. Ren, Homer Walke, Quan Vuong, Lucy Xiaoyang Shi, and Sergey Levine. Knowledge insulating vision-language-action models: Train fast, run fast, generalize better. In *Advances in Neural Information Processing Systems*, volume 38, 2025.
- [15] Federica Ferraguti, Cristian Secchi, and Cesare Fantuzzi. A tank-based approach to impedance control with variable stiffness. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.
- [16] Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.
- [17] Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning (ICML)*, 2018.
- [18] Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, et al. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*, 2025.
- [19] Zihao He, Hongjie Fang, Jingjing Chen, Hao-Shu Fang, and Cewu Lu. Foar: Force-aware reactive policy for contact-rich robotic manipulation. *IEEE Robotics and Automation Letters*, 10(6):5625–5632, 2025.
- [20] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS Workshop on Deep Generative Models and Downstream Applications*, 2021. *arXiv:2207.12598*.
- [21] Neville Hogan. Impedance control: An approach to manipulation, parts i–iii. *ASME Journal of Dynamic Systems, Measurement, and Control*, 107(1):1–24, 1985.
- [22] Jialei Huang, Shuo Wang, Fanqi Lin, Yihang Hu, Chuan Wen, and Yang Gao. Tactile-VLA: Unlocking vision-language-action model’s physical knowledge for tactile generalization. *arXiv preprint arXiv:2507.09160*, 2025.
- [23] Tadanobu Inoue, Giovanni De Magistris, Asim Munawar, Tsuyoshi Yokoya, and Ryuki Tachibana. Deep reinforcement learning for high precision assembly tasks. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.
- [24] Tobias Johannink, Shikhar Bahl, Ashvin Nair, Jianlan Luo, Avinash Kumar, Matthias Loskyll, Juan Aparicio Ojea, Eugen Solowjow, and Sergey Levine. Residual reinforcement learning for robot control. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

- [25] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, et al. Droid: A large-scale in-the-wild robot manipulation dataset. In *Robotics: Science and Systems (RSS)*, 2024.
- [26] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P. Foster, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 2679–2713. PMLR, 2025.
- [27] Klas Kronander and Aude Billard. Stability considerations for variable impedance control. *IEEE Transactions on Robotics*, 32(5):1298–1305, 2016.
- [28] Kun Lei, Huanyu Li, Dongjie Yu, Zhenyu Wei, Lingxiao Guo, Zhennan Jiang, Ziyu Wang, Shiyu Liang, and Huazhe Xu. RL-100: Performant robotic manipulation with real-world reinforcement learning. *arXiv preprint arXiv:2510.14830*, 2025.
- [29] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.
- [30] Guanxing Lu, Wenkai Guo, Chubin Zhang, Yuheng Zhou, Haonan Jiang, Zifeng Gao, Yansong Tang, and Ziwei Wang. VLA-RL: Towards masterful and general robotic manipulation with scalable reinforcement learning. *arXiv preprint arXiv:2505.18719*, 2025.
- [31] Jianlan Luo, Zheyuan Hu, Charles Xu, You Liang Tan, Jacob Berg, Archit Sharma, Stefan Schaal, Chelsea Finn, Abhishek Gupta, and Sergey Levine. Serl: A software suite for sample-efficient robotic reinforcement learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.
- [32] Jianlan Luo, Charles Xu, Fangchen Liu, Liam Tan, Zipeng Lin, Jeffrey Wu, Pieter Abbeel, and Sergey Levine. FMB: A functional manipulation benchmark for generalizable robotic learning. *International Journal of Robotics Research*, 44(4):592–606, 2025.
- [33] Jianlan Luo, Charles Xu, Jeffrey Wu, and Sergey Levine. Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning. *arXiv preprint arXiv:2410.21845*, 2024.
- [34] Matthew T. Mason. Compliance and force control for computer controlled manipulators. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(6):418–432, 1981.
- [35] Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. RoboCasa: Large-scale simulation of household tasks for generalist robots. In *Robotics: Science and Systems*, 2024.
- [36] NVIDIA. NVIDIA Isaac GR00T N1.7: A foundation model for generalist robots. <https://github.com/NVIDIA/Isaac-GR00T/releases/tag/n1.7-release>, 2026. Official N1.7 release, accessed 2026-08-28.
- [37] NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.
- [38] Open X-Embodiment Collaboration, Abby O’Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, et al. Open x-embodiment: Robotic learning datasets and rt-x models. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.
- [39] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.
- [40] 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. In *Proceedings of Robotics: Science and Systems*, 2025.
- [41] Physical Intelligence. $\pi_{0,6}^*$: A vla that learns from experience. *Technical report*, 2025.

- [42] Marc H. Raibert and John J. Craig. Hybrid position/force control of manipulators. *ASME Journal of Dynamic Systems, Measurement, and Control*, 103(2):126–133, 1981.
- [43] Tom Silver, Kelsey Allen, Josh Tenenbaum, and Leslie Kaelbling. Residual policy learning. *arXiv preprint arXiv:1812.06298*, 2018.
- [44] Daniel Sliwowski, Shail Jadav, Sergej Stanovcic, Jędrzej Orbik, Johannes Heidersberger, and Dongheui Lee. REASSEMBLE: A multimodal dataset for contact-rich robotic assembly and disassembly. *arXiv preprint arXiv:2502.05086*, 2025.
- [45] Markku Suomalainen, Yiannis Karayiannidis, and Ville Kyrki. A survey of robot manipulation in contact. *Robotics and Autonomous Systems*, 156:104224, 2022.
- [46] Luigi Villani and Joris De Schutter. Force control. In *Springer Handbook of Robotics*, pages 195–220. Springer, 2nd edition, 2016.
- [47] Andrew Wagenmaker, Yunchu Zhang, Mitsuhiro Nakamoto, Seohong Park, Waleed Yagoub, Anusha Nagabandi, Abhishek Gupta, and Sergey Levine. Steering your diffusion policy with latent space reinforcement learning. In *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pages 258–282. PMLR, 2025.
- [48] Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, Zhuqin Yang, Meng Li, Yinuo Zhao, Zhiyuan Xu, Guang Yang, et al. RoboMIND: Benchmark on multi-embodiment intelligence normative data for robot manipulation. In *Proceedings of Robotics: Science and Systems*, 2025.
- [49] Jiawen Yu, Hairuo Liu, Qiaojun Yu, Jieji Ren, Ce Hao, Haitong Ding, Guangyu Huang, Guofan Huang, Yan Song, Panpan Cai, Cewu Lu, and Wenqiang Zhang. ForceVLA: Enhancing VLA models with a force-aware MoE for contact-rich manipulation. *Advances in Neural Information Processing Systems*, 2025. arXiv:2505.22159.
- [50] Xiu Yuan, Tongzhou Mu, Stone Tao, Yunhao Fang, Mengke Zhang, and Hao Su. Policy decorator: Model-agnostic online refinement for large policy model. In *International Conference on Learning Representations*, 2025.
- [51] Shuoheng Zhang, Yifu Yuan, Hongyao Tang, Yan Zheng, Qiaojun Yu, Pengyi Li, Guowei Huang, Helong Huang, Xingyue Quan, and Jianye Hao. ForceFlow: Learning to feel and act via contact-driven flow matching. *arXiv preprint arXiv:2605.11048*, 2026.
- [52] Zongzheng Zhang, Haobo Xu, Zhuo Yang, Chenghao Yue, Zehao Lin, Huan-ang Gao, Ziwei Wang, and Hao Zhao. Elucidating the design space of torque-aware vision-language-action models. In *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pages 4019–4037. PMLR, 2025.
- [53] Ruiteng Zhao, Wenshuo Wang, Yicheng Ma, Xiaocong Li, Francis E. H. Tay, Marcelo H. Ang, and Haiyue Zhu. FD-VLA: Force-distilled vision-language-action model for contact-rich manipulation. *arXiv preprint arXiv:2602.02142*, 2026.

A Assembly Suite Details

The suite uses a shared chassis fixture but exposes distinct contact geometries. Fig. 4 in the body records each execution chain; Table 5 gives the evaluated clearances and configured force limits. LEVER is evaluated as an isolated, contact-only CPU socket lever closure rather than as another full pick-to-place chain.

Table 5. The host-computer assembly suite: 23 sub-tasks, fourteen of them contact-critical. Sub-goals are listed in execution order. Clearance is the tightest fit among the task’s critical sub-goals. $F_{\max}$ is the configured safety limit for the indicated contact phase; CPU uses 8 N for pin-array seating and 35 N for socket closure.

Task	Sub-goal sequence (free-space, contact-critical)	Clear. (mm)	$F_{\max}$ (N)	Contact character
RAM install	pick, hold, align, place, press	0.18	70	sequential per-end latch presses
CPU install	pick, hold, rotate, align, place, push, press	0.10	8 / 35	fragile pin-array seating; ZIF closure
Disk install	pick, hold, align, place	0.30	20	guided bay seating
GPU install	pick, rotate, align, place, press	0.18	55	long-edge PCIe alignment; latch click
LEVER closure	press, press	0.10	60	two sequential lever presses

A.1 Evaluation Protocol

This subsection restores the setup detail compressed out of Sec. 6.1.

Task suite. Each task carries an overall instruction, for example *pick up the RAM and install it*, which the backbone decomposes into the sub-goals shown in Fig. 4, for example *press down the RAM until it locks into place*. The five tasks comprise 23 sub-tasks drawn from a shared seven-verb vocabulary (*pick, hold, rotate, align, place, push, press*), so a skill learned in one task is addressable in another. Nine sub-tasks are free-space (*pick, hold, rotate*) and execute on the coarse chunk alone; the remaining fourteen are contact-critical (*align, place, push, press*) and activate the refinement expert. Clearances range from 0.10 to 0.30 mm, and each contact phase uses a configured safety limit $F_{\max}$; Fig. 4 and Table 5 give the full sequences, clearances and limits.

Protocol. Chassis pose is randomized per trial and parts are supplied by an automated feeder. Each cell in the main tables is 20 trials under a protocol declared before evaluation, with no re-runs.

Metrics. *Success rate*: the installation completed without force violation. *Peak off-axis force*: the maximum contact force orthogonal to the task-frame motion axis, which measures misalignment damage risk rather than deliberate seating force. *Violation rate*: the fraction of trials with any control step above $F_{\max}$. *Intervention rate*: the fraction of trials in which a human supervisor took over. *Recovery rate*: the fraction of detected failures the system resolved on its own. *Cycle time*: wall-clock per successful trial. *Inference latency*: wall-clock from observation to command.

Baselines. All baselines are trained on ManuFacet-1K with matched budgets and evaluated under one aligned protocol. $\pi_{0.5}$ [7] uses the same vision–language observations and 7-D kinematic state but no wrench channel. $\pi_{0.5}+F$ adds the wrench as an input token and changes nothing else, which isolates force as an input from force as a lifecycle commitment. $\pi_{0.5}+\text{RECAP}$ -style applies the published RECAP value-estimation and advantage-conditioning objective [41] to the same $\pi_{0.5}$ backbone; it is our aligned reimplementaion rather than the released $\pi_{0.6}^*$ model. GR00T N1.7 [36] is a second generalist backbone of comparable scale. TA-VLA [52] is an aligned reimplementaion on the same backbone that aggregates wrist F/T history into a single token. The *Align* and *+RL* variants of Facet-0 denote semantic–contact alignment alone and its value-guided refinement, respectively; *Full* additionally includes local policy adaptation.

B Dataset Curation Protocol

This section expands the collection and curation summary of Sec. 3. The acquisition network uses a shared sensing specification: each cell has a torque-controlled arm, parallel gripper, wrist six-axis force/torque sensor, three RGB cameras, and joint proprioception. Kinesthetic teleoperation with force feedback is used for demonstrations. Hardware timestamps on a shared clock define a common 15 Hz training timeline;

high-rate state and wrench streams are retained for the 200 Hz controller. Episodes whose cross-channel skew exceeds tolerance are rejected rather than interpolated.

Validity extraction. Teleoperation faults, truncated chunks, sensor saturation, calibration drift beyond bounds, and sustained manipulated-part occlusion are removed at segment level, preserving unaffected intervals of an episode. Action chunks are scored by action and wrench variation over the horizon; near-stationary pauses, repositioning and fixture waits are down-sampled to avoid duration-dominated mixtures.

Semantic and phase annotation. Each segment receives two VLM-generated question–answer pairs: its overall task/current sub-task and the next instruction (e.g., “press down the RAM until it locks into place”). Annotators mark boundaries for the seven phases (*approach, align, insert, press, seat, fasten, retreat*) using keyboard chords; the VLM pass verifies boundaries and flags disagreements for re-annotation. The resulting labels are reused verbatim by pretraining and reward decomposition.

Anomaly reflow. Trajectories with anomalous jumps are paired with the corrected trajectory that follows and both are retained. This preserves failure and recovery signal for post-training while allowing the curation log to trace each correction. Center identities are pseudonymised, personally identifying content is removed, and the release includes the data card and curation code.

C Architecture and Execution Details

This appendix records the engineering realization of the architecture of Sec. 4.1: the execution hierarchy, the compliant controller, the safety operator, and the design rationale compressed out of the main text. Nothing here changes the model definition.

Why the wrench history is a window. A single wrench sample is ambiguous. Contact onset is a *step* in the wrench, a jam is a wrench rising while displacement stalls, and a skew is an off-axis moment persisting under axial push; each is a property of a short trajectory, not of one frame, and none is visible to vision once the part occludes the slot. $K = 10$ frames is the shortest window over which the contact-onset step is detectable above sensor noise at the corpus sampling rate.

Why the wrench is predicted rather than only read. Conditioning an action on the current wrench hands the policy a record of contact its own past actions already produced: it arrives too late to shape the action that caused it, and nothing in a behavior-cloning loss rewards attending to six force dimensions among thousands of visual ones. Making the wrench a decoding target changes the learning signal: the wrench term in Eq. (3) falls only for a model that represents the contact consequence of its proposed motion, and $\hat{W}$ becomes a predictive variable available before execution. Fig. 9 schematically summarizes the intended roles of this variable across training.

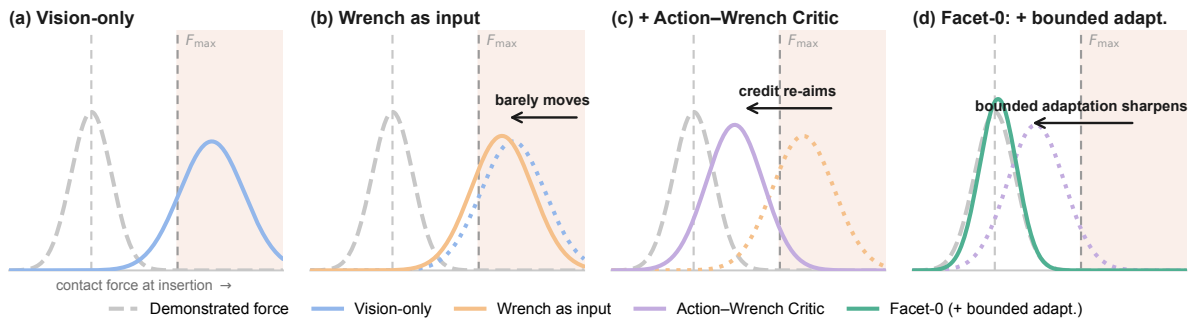


Figure 9. Schematic of how the action–wrench representation is intended to shape contact behavior across optimization. The distributions illustrate the roles of input conditioning, Action–Wrench valuation, and bounded local adaptation; they are conceptual rather than empirical measurements.

Execution hierarchy. The joint action–wrench proposal of Sec. 4.1 is decoded at 5–10 Hz at roughly 5 mm resolution. A contact-rate refinement expert consumes that chunk together with h_t^c and returns a refined action–wrench chunk at 20 Hz at roughly 0.5 mm; coarse and fine differ in precision, not in kind.

Beneath both, a compliant controller runs at 200 Hz. The refinement expert is frozen during the local policy adaptation of Sec. 5.3.

Compliant controller. The 200 Hz layer regulates force along the task-frame directions designated by the sub-goal and tracks position along the remainder [5, 21, 34, 42]. Its force loop closes on measured wrist F/T feedback. Predicted wrench is retained as a supervised contact-consequence variable for representation and value learning; it is neither a controller input nor a force setpoint.

Safety operator. $\mathcal{S}_{\text{task}}$ sits between policy and controller and clips the commanded target to the workspace, limits per-step motion, canonicalizes or passes through rotation, optionally smooths in $SE(3)$, gates spatially, and passes the gripper command through unchanged. It is architectural rather than behavioral: a policy *trained* to be gentle is gentle on its training distribution, whereas a clipped and rate-limited command is constrained on every distribution, including those that on-robot adaptation, mis-timed sub-goal boundaries, and unseen parts produce. Its guarantee is kinematic and we state its limit plainly: bounding displacement per step bounds how far a mis-aimed target can drive the tool between observations, which is what keeps early adaptation episodes cheap, but it is not the passivity bound of energy-tank formulations [15, 27]. The reward and compliant controller discourage excessive interaction force, but no certified hard force bound is claimed.

Why adapt behind an interface rather than fine-tune the backbone. Three reasons, in increasing order of importance. It is cheap: adaptation to an unseen part touches only the local actor and finishes in hours on one robot. It is safe in the regression sense: the backbone carrying open-world semantics is never updated by a few hundred contact episodes, so adding a skill cannot silently degrade the others. And it is safe in the operational sense: the actor’s output is confined to the per-task admissible box $[a_{\min}, a_{\max}]$ of Sec. 5.3 and then passed through $\mathcal{S}_{\text{task}}$, so every command is admissible by construction regardless of how far the actor departs from the frozen reference. We adopt a bounded absolute parameterization rather than a residual one precisely because a residual’s safety comes from the smallness of its corrections, and that same smallness bounds departure from a base whose failures adaptation exists to repair.

Predictive coupling across optimization. Semantic–contact alignment and value-guided refinement operate on the joint action–wrench proposal Y_t introduced in Sec. 4.1. Local policy adaptation retains the same predictive relation through a bounded action head and a supervised one-step wrench head. Its twin critics and robot interface remain defined only over the executable Cartesian action; the wrench prediction is auxiliary and non-commanded.

D Method Details

This appendix collects the optimization procedures, constants, and derivations deferred from Secs. 4 and 5. The three method modules share the semantic–contact representation h_t^c and preserve its predictive action–contact relation. Semantic–contact alignment learns h_t^c and decodes the joint proposal $Y_t = [A_{t:t+H-1}, \widehat{W}_{t+1:t+H}]$ from demonstrations; value-guided policy refinement evaluates that proposal with the Action–Wrench Critic; and local policy adaptation compresses h_t^c into z_t while adapting only the executable action to a new part (Table 6). Throughout, Ω denotes the VLM judge and $F_{\max}(\varphi)$ the force limit configured for contact phase φ .

Table 6. The three method modules share h_t^c while retaining predictive coupling between action and contact consequence. Y_t denotes the decoded joint proposal, not the representation itself.

Module	Representation / proposal role	Trainable	Frozen	Data
Semantic–contact alignment	learns h_t^c ; decodes Y_t	π_{align}	–	ManuFacet-1K demos
Value-guided refinement	values Y_t ; retains h_t^c	$\pi_{\text{ref}}, Q_{\psi}^{\text{AW}}, E_z$	VL backbone	rollouts + interventions
Local policy adaptation	compresses h_t^c ; bounded heads	local actor η, Q_{ξ}	$\pi_{\text{ref}}, E_z, \Omega$	on-robot experience

D.1 Semantic–Contact Alignment

Masking and normalization. M^a and M^w mask padded and invalid timesteps in the action and wrench slices of Y_t , respectively. Each term of Eq. (3) is normalized by its own mask sum rather than by a shared denominator, so a batch with few valid contact frames does not silently down-weight the force term.

Optimizer alternation and update budget. Algorithm 1 selects the action–wrench branch on four updates in five, giving 32K action-branch updates against 8K VQA updates on one run.

Why the losses are never summed. $\mathcal{L}_{AW}$ is a flow-matching MSE and $\mathcal{L}_{VQA}$ a token cross-entropy; they carry different units and differ by orders of magnitude, so a single weighted sum would make the effective weighting an artifact of scale rather than a design choice. Alternating two optimizers four-to-one keeps each branch’s step size independent.

Mixture pyramid. The three tiers are fundamental operations (*pick, hold, rotate*), precision operations (*align, place, push, press*), and long-tail recovery episodes. Sampling weights over-represent the upper two tiers relative to their raw duration, because duration and decisiveness are inversely related in assembly: free-space transit dominates wall-clock time and decides nothing.

Algorithm 1 Semantic–contact alignment (Eq. (3))

Require: demonstrations $\mathcal{D}_{pre}$, VQA set $\mathcal{D}_{vqa}$, pretrained VL backbone, mixture weights over {base, precision, recovery}

Ensure: aligned policy π_{align} generating $Y_t = [A_{t:t+H-1}, \hat{W}_{t+1:t+H}] \in \mathbb{R}^{H \times 13}$

```

1: for update  $k = 0, 1, 2, \dots$  do
2:   if  $k \bmod 5 \in \{0, 1, 2, 3\}$  then ▷ action–wrench branch
3:     sample  $(I_t^{1:3}, \ell, s_t^{kin}, W_{t-K+1:t}, Y_t^{data})$  from  $\mathcal{D}_{pre}$  under the mixture pyramid
4:      $h_t^{VL} \leftarrow E_{VL}(I_t^{1:3}, \ell); \quad h_t^c \leftarrow F_{fuse}(h_t^{VL}, E_s(s_t^{kin}), E_w(W_{t-K+1:t}))$  ▷ Eq. (2)
5:      $\epsilon \sim \mathcal{N}(0, I); \quad \tau \sim \text{Beta}(1.5, 1)$ 
6:      $X_\tau \leftarrow (1 - \tau)Y_t^{data} + \tau\epsilon; \quad U_\tau \leftarrow \epsilon - Y_t^{data}$ 
7:      $v \leftarrow v_\theta(X_\tau, \tau \mid h_t^c)$ 
8:      $\mathcal{L}_{act} \leftarrow \sum M^a \odot \|v^a - U_\tau^a\|_2^2 / \sum M^a$  ▷ action columns
9:      $\mathcal{L}_{wr} \leftarrow \sum M^w \odot \|v^w - U_\tau^w\|_2^2 / \sum M^w$  ▷ wrench columns
10:     $\theta \leftarrow \text{Opt}_{AW}(\theta, \nabla \mathcal{L}_{AW}), \quad \mathcal{L}_{AW} = \mathcal{L}_{act} + \lambda_{pre} \mathcal{L}_{wr}$  ▷  $\lambda_{pre} = 0.1$ 
11:   else ▷ VQA branch
12:     sample  $(I, q, y_{1:L})$  from  $\mathcal{D}_{vqa}$ 
13:      $\mathcal{L}_{VQA} \leftarrow -(\sum_i m_i)^{-1} \sum_i m_i \log p_{\theta_{VLM}}(y_i \mid I, q, y_{<i})$ 
14:      $\theta \leftarrow \text{Opt}_{VQA}(\theta, \nabla \mathcal{L}_{VQA})$ 
15:   end if
16: end for
```

Alignment contract across the two branches. Alternating optimization is useful only if the branches improve a common representation rather than exchange targets. The structured mask therefore exposes the same visual–language prefix to both branches while keeping their supervised outputs disjoint. The action–wrench branch additionally receives kinematic state, causal wrench history, and the noised joint chunk; the VQA branch receives preceding answer tokens. Neither branch can attend to the other’s target tokens. Semantic supervision can consequently shape the shared prefix without revealing the answer sequence to the policy, while interaction supervision can attach physical consequence to that prefix without turning wrench into a language label.

Temporal pairing and causality. Row k of Y_t^{data} pairs the command at $t + k$ with the measured wrench at $t + k + 1$. This one-step offset is the smallest causal convention that assigns contact to the motion that produced it. Pairing an action with the contemporaneous pre-action sample would instead teach the decoder to reconstruct contact already present in the observation. Invalid or padded future samples are removed by M^w , so the offset changes neither the action horizon nor the number of executable commands. The resulting representation summarizes what the scene means, what the robot is doing, and what contact that motion is expected to produce.

D.2 Value-Guided Policy Refinement

Algorithm 2 Value-guided policy refinement with the Action–Wrench Critic

Require: aligned policy π_{align} , task set $\mathcal{T}$, VLM judge Ω

Ensure: value-refined policy π_{ref} , Action–Wrench Critic Q_{ψ}^{AW} , bottleneck encoder E_z

- 1: initialize $\pi_{\text{ref}} \leftarrow \pi_{\text{align}}$
 - 2: **repeat**
 - 3: $\mathcal{D} \leftarrow \mathcal{D} \cup \text{rollout}(\pi_{\text{ref}}, \mathcal{T}) \cup \text{supervisor interventions}$
 - 4: $r_t \leftarrow \text{Eq. (4) under } \Omega\text{'s phase segmentation; } c_t \leftarrow \text{smoothed contact intensity}$
 - 5: train Q_{ψ}^{AW} by Eq. (5): distributional TD, progress-calibrated, four aux heads
 - 6: $\delta_t^{(N)} \leftarrow \text{Eq. (6); } b_t \leftarrow \text{within-regime threshold}$ ▷ contact-enriched
 - 7: retain $Y_t^{\text{data}} = [A_{t:t+H-1}^{\text{data}}, W_{t+1:t+H}^{\text{data}}]$ with contact-validity masking
 - 8: flow-matching update of π_{ref} by Eq. (7): frames with $b_t = 1$ train the conditional branch, the rest the unconditional branch
 - 9: **until** round budget exhausted
 - 10: train bottleneck encoder E_z on frozen backbone embeddings ▷ used for local policy adaptation
-

Distributional parameterization and auxiliary heads. Z_{ψ} of Eq. (5) is a distributional return model [4] trained on all rollouts under progress-calibrated targets. The concrete distributional parameterization and regression distance are implementation-dependent; the method requires only a return distribution whose mean defines Q_{ψ} . Four auxiliary heads share the critic trunk and are trained jointly with the return objective: near-future wrench, contact intensity c_t , within-contact progress, and success ranking. They exist to separate observations that agree on task progress but disagree on contact state; without that separation the credit score of Eq. (6) cannot express a preference between gentle alignment and ramming, because at equal progress the two are the same point.

Wrench supervision during refinement. Value-guided refinement does not introduce a new wrench variable. It retains the action-to-next-wrench pairing learned during semantic–contact alignment: each action row is supervised by the measured wrist wrench at the following step. Causal filtering, per-axis normalization, and a contact-validity mask are applied only to stabilize the loss; after de-normalization, $\tilde{W}_{t+1:t+H}$ retains the physical interpretation given in Sec. 4.1. The executable action columns are unchanged.

Contact regime and positive-label budget. The regime ρ_t of Eq. (7) is obtained by thresholding the contact intensity c_t , and q_{ρ} is the score threshold taken *within* regime ρ at a matched overall positive rate. Ranking within regime rather than globally prevents the positive set from being dominated by free-space progress.

Why the credit score is undiscounted and short. Under a progress-calibrated critic, V_{ψ} is approximately affine in normalized task progress. A discounted bootstrap over the full chunk horizon therefore makes the score affine in $V_{\psi}(h_t^c)$ itself, so early frames of every successful trajectory score highly merely because they are early, and the positive-label budget collects on the free-space approach the policy already performs well. Truncating to $N < H$ and dropping the discount makes $\delta_t^{(N)}$ measure local deviation from the trajectory’s own calibrated progress rate, which is invariant to where in the trajectory the frame sits.

Prompt-level conditioning. The positive-interaction label b_t is appended to the instruction as a short literal tag, so the conditional and unconditional branches of Eq. (7) differ only in that suffix and no architectural change is required. At inference the two combine in classifier-free-guidance style [20]; the guidance scale is a deployment dial and $s = 0$ recovers the unconditional policy exactly. Branch ratios are set so the unconditional branch always sees the full data distribution and continues to act as a behavior-cloning anchor.

Contact intensity. $c_t \in [0, 1]$ is computed from causally smoothed wrench differences rather than raw per-step deltas. Raw deltas in the high-rate sensor stream are dominated by noise; the smoothing window is the shortest that makes the contact-onset step detectable above sensor noise.

D.3 Local Policy Adaptation

FACET bottleneck. On task data, E_z learns a frozen, per-valid-token reconstruction bottleneck over h_t^c . Its output, the FACET token $z_t = E_z(h_t^c)$, lets local policy adaptation consume the aligned scene state without backpropagating into the backbone.

Actor objective. The learner is a TD3-style deterministic actor-critic [17] with twin critics and target networks. The critics are defined on the executable Cartesian action and regress onto the command produced by $\mathcal{S}_{\text{task}}$. Successful human interventions provide the action anchor [16, 33], while realized next-step wrench supervises the auxiliary consequence head:

$$\begin{aligned} \mathcal{L}_{\text{actor}}(\eta) = & -\mathbb{E}_{\text{on}}[Q_{\xi_1}(e_t, a_t)] \\ & + \lambda_{\text{BC}} \mathbb{E}_{\text{BC}} \left[\|M_{\mathcal{T}} \odot (\bar{a}_t - \bar{a}_t^{\text{human}})\|_2^2 \right] \\ & + \lambda_{\text{pred}} \mathbb{E}_{\text{on}}[\mathcal{L}_w(\hat{w}_{t+1}, w_{t+1})]. \end{aligned} \quad (9)$$

Here $\mathbb{E}_{\text{on}}$ and $\mathbb{E}_{\text{BC}}$ average over online transitions and successful intervention samples, respectively; $\bar{a}_t$ and $\bar{a}_t^{\text{human}}$ are the normalized actor and intervention actions; $M_{\mathcal{T}}$ masks dimensions outside the task action space; and λ_{BC} and λ_{pred} weight action anchoring and wrench prediction.

Deployment interface. The deployed actor retains both heads, but only $e_t \rightarrow f_{\eta} \rightarrow a_t \rightarrow \mathcal{S}_{\text{task}} \rightarrow \text{robot}$ is executable. The actor input concatenates the FACET token z_t , the normalized proprioceptive vector $[x_t; g_t; w_t]$ of Eq. (1), and the frozen reference $\tilde{a}_t$. The wrench head remains fully supervised, predictive, and non-commanded.

Feature construction. The four modality embeddings are

$$e^z = \text{LN}(W_z z_t), \quad e^p = \text{LN}(W_p [x_t, g_t]), \quad e^w = \text{LN}(W_w w_t), \quad e^{\text{ref}} = \text{LN}(W_a \tilde{a}_t),$$

where LN is layer normalization and W_z , W_p , W_w , and W_a are learned modality projections. Each embedding is in $\mathbb{R}^{256}$, and their concatenation forms the actor input $e_t \in \mathbb{R}^{1024}$ of Sec. 5.3. A shared MLP actor f_{η} produces $[u_t^a, \hat{w}_{t+1}]$: the raw action activation $u_t^a \in \mathbb{R}^7$ is squashed into the per-task box $[a_{\min}, a_{\max}]$ to form a_t , and $\hat{w}_{t+1} \in \mathbb{R}^6$ is supervised by the measured next-step wrench through Eq. (9). The twin critics consume only a_t as the executable action and do not treat $\hat{w}_{t+1}$ as an action dimension.

Why the critic scores the applied action. $\mathcal{S}_{\text{task}}$ can alter a_t before execution, so replay stores the safety-filtered value in the same action field. This convention keeps the critic’s regression target consistent with the observed dynamics without introducing a second action symbol.

Why behavior cloning is success-filtered. An intervention records what a human did, not that it worked, so promoting every intervention would anchor the actor to recoveries that themselves failed. Filtering on later confirmation keeps only segments known to reach the sub-goal, which matters most early in adaptation when the RL term is still driven by sparse, noisy reward. The confirmation key never enters the observation, so the policy cannot learn to read it.

Sample-efficiency details. Two choices support on-robot sample efficiency. *Single-step commands:* the actor emits one absolute Cartesian target per decision rather than a chunk of horizon H , so every environment step yields a transition and the credit assignment horizon stays short. *Bounded output:* the tanh squashing into per-task limits $[a_{\min}, a_{\max}]$ (Sec. 5.3) replaces the small-correction assumption that residual baselines rely on: the actor may propose targets far from the reference, but never outside the task’s admissible box.

Representation-to-policy interface. The FACET token and the measured wrench play complementary roles at the local actor. The token carries task intent and the contact structure learned from the larger corpus, whereas w_t reports the interaction currently unfolding on the new part. Conditioning on both allows the same aligned scene state to support different bounded targets when measured contact differs. The frozen reference supplies the nominal task motion, while the actor learns only the part-dependent correction expressed as an absolute target.

Algorithm 3 Bounded local policy adaptation with an auxiliary wrench head

Require: frozen $\{\pi_{\text{ref}}, E_z, \Omega\}$; actor f_η ; twin action critics Q_{ξ_i} ; bounds $a_{\min}, a_{\max}$; mask $M_{\mathcal{T}}$; operator $\mathcal{S}_{\text{task}}$

Ensure: bounded action policy and supervised auxiliary wrench predictor

```
1: loop
2:    $z_t \leftarrow E_z(h_t^c)$  ▷ FACET token from the aligned representation
3:    $\tilde{a}_t \leftarrow$  reference action from frozen  $\pi_{\text{ref}}$ ;  $e_t \leftarrow [z_t; \text{Norm}(s_t^{\text{kin}}, w_t); \tilde{a}_t]$  ▷ conditioning, not a base
4:    $[u_t^a, \hat{w}_{t+1}] \leftarrow f_\eta(e_t)$ ;  $a_t \leftarrow a_{\min} + \frac{1}{2}(\tanh u_t^a + 1) \odot (a_{\max} - a_{\min})$  ▷ Sec. 5.3
5:    $a_t \leftarrow \mathcal{S}_{\text{task}}(a_t)$ ; execute  $a_t$  ▷ wrench prediction is non-commanded
6:    $\mathcal{D}_{\text{online}} \leftarrow \mathcal{D}_{\text{online}} \cup \{(e_t, a_t, r_t, d_t, e_{t+1}, w_{t+1})\}$ 
7:   if human intervened and the segment is later confirmed successful then
8:      $\mathcal{D}_{\text{BC}} \leftarrow \mathcal{D}_{\text{BC}} \cup \{(e_t, \tilde{a}_t, a_{\text{human}, t})\}$ 
9:   end if
10:   $\hat{q}_t \leftarrow r_t + \gamma(1 - d_t) \min_i Q_{\xi_i}(e_{t+1}, \pi_{\tilde{\eta}}^a(e_{t+1}))$ 
11:   $\xi_i \leftarrow \xi_i - \alpha_Q \nabla_{\xi_i} \mathbb{E}[(Q_{\xi_i}(e_t, a_t) - \hat{q}_t)^2]$ 
12:   $\eta \leftarrow \eta - \alpha_\pi \nabla_\eta \mathcal{L}_{\text{actor}}$  ▷ Eq. (9)
13: end loop
```

Here $\tilde{\xi}_i$ and $\tilde{\eta}$ denote target-critic and target-actor parameters, $\pi_{\tilde{\eta}}^a$ is the action head of the target actor, $\hat{q}_t$ is its one-step TD target, and α_Q and α_π are the critic and actor learning rates.

Separation of reference, command, and consequence. Algorithm 3 keeps three quantities distinct. The frozen action $\tilde{a}_t$ is a reference that informs the actor but is never added as a residual base. The bounded output a_t is the proposed Cartesian target, and the value stored in replay is the target after the task safety operator has been applied. The auxiliary $\hat{w}_{t+1}$ predicts the measured consequence of that applied motion. This separation lets the critic remain a value function over the executable action space while the shared actor trunk retains an explicit model of how a correction changes contact.

Frozen and adaptable information. The semantic–contact encoder, FACET bottleneck, reference policy, VLM judge, compliant controller, and safety operator remain fixed. Only the lightweight actor and its twin critics adapt to the new part. The boundary is intentional: language semantics, scene interpretation, and coarse task progress are reusable across parts, whereas local mass, stiffness, latch response, and contact geometry determine which bounded correction succeeds. Freezing the upstream representation prevents a small on-robot dataset from rewriting those reusable quantities, while conditioning on the reference keeps the adapted actor anchored to the task-level behavior learned from the larger corpus.

Asymmetric use of the two actor heads. Both heads remain active throughout adaptation, but they enter optimization differently. The action head is evaluated by TD3 and is the only head connected to execution. The wrench head is trained only against future wrist F/T and regularizes the shared trunk toward action–contact consistency. It neither enlarges the Markov decision process action nor closes the robot’s force loop; measured feedback remains the input to compliant control. The local update therefore changes how the policy reacts to contact without introducing a second force-command interface or changing the safety contract used by the rest of the system.

Consistency under safety filtering. The transition stored in replay contains the command that actually reaches the controller, not the actor’s pre-filter proposal. This distinction matters whenever workspace clipping or rate limits alter a target: the observed reward and next state were caused by the applied command, so regressing the critic on the unapplied proposal would assign value to an action the robot never took. The actor can still improve its bounded proposal through the critic, but every target used for value learning remains grounded in the closed-loop trajectory. The safety operator therefore stays external to the learned policy without creating an inconsistency between optimization and execution.

E Reported Per-Sub-Goal Results

Table 7 pairs the five task-level results of Table 2 with a reported subset of 17 of the suite’s 23 sub-goals, including 13 contact-critical rows. The subset is visualized for representative methods in Fig. 5 and diagnoses whether failures concentrate at contact; it is not presented as a complete 23-row decomposition. Aggregate contact means are unweighted over the 13 shown contact rows and rounded to the nearest point. The *Align*, *+RL*, and *Full* columns match the variant names used in the body.

Table 7. Reported task and sub-goal success rates (%), 20 trials per cell. The 17 displayed sub-goals are a diagnostic subset of the 23-sub-goal suite; † marks the 13 contact-critical rows. Shaded rows report task-level chain success.

Sub-goal	Baselines					Facet-0 (ours)		
	$\pi_{0.5}$	+F	+RECAP	GR00T	TA-VLA	Align	+RL	Full
<i>RAM installation</i>								
pick	100	100	100	95	100	100	100	100
align†	10	15	35	10	10	15	45	95
place†	40	65	70	40	60	65	75	100
press†	80	85	95	75	80	85	100	95
RAM task	10	15	35	10	10	15	45	95
<i>CPU installation</i>								
pick	80	80	100	80	100	100	100	100
align†	5	5	15	5	20	20	45	85
place†	55	60	65	55	60	65	75	95
press†	70	70	70	65	70	75	80	95
CPU task	5	5	15	5	20	20	45	85
<i>Disk installation</i>								
pick	65	70	65	60	90	100	100	100
align†	25	20	20	5	30	30	65	95
place†	70	70	75	75	90	90	100	100
Disk task	25	20	20	5	30	30	65	95
<i>GPU installation</i>								
pick	70	70	70	75	80	100	100	100
align†	10	5	5	0	5	10	35	85
place†	25	25	30	25	25	30	85	95
press†	20	20	25	25	60	60	65	90
GPU task	10	5	5	0	5	10	35	85
<i>LEVER closure</i>								
press† (first)	0	0	0	0	5	5	0	50
press† (second)	0	0	0	0	5	5	0	55
LEVER task	0	0	0	0	5	5	0	50
Mean, 13 reported contact sub-goals†	32	34	39	29	40	43	59	87
Mean, five tasks	10	9	15	4	14	16	38	82

Reading the decomposition. Task rows require completion of the full chain, whereas sub-goal rows isolate individual decision points. For the full system, each chain rate matches its weakest displayed contact-critical sub-goal: alignment limits RAM, CPU, Disk, and GPU, while the first press limits LEVER. The 13-row contact mean therefore measures local interaction quality with equal weight per event; the five-task mean measures end-to-end reliability with equal weight per assembly. Reporting both separates contact quality from chain completion without allowing longer tasks to dominate the diagnostic average.

F Supporting Evaluation Data

This section retains the exact reference values behind the compact visualizations in the main text, together with the deployment-properties table moved out of the main narrative. On the same fixtures, Facet-0 completes the task 40% faster than expert teleoperation.

Table 8. Deployment efficiency and precision; placement accuracy is measured at the align-to-insert transition.

Method	Inference latency (ms) ↓	Placement accuracy (mm) ↓
$\pi_{0.5}$ [7]	150	≈ 5
Facet-0 (Full)	50	0.5

Table 9. Disk post-training results underlying Fig. 6, with matched deployment data, VLM reward, and budget.

Metric	Matched AWR	Facet-0 post-training
Success rate ↑	20%	65%
Intervention rate ↓	47%	24%
Recovery rate ↑	44%	81%

Table 10. Adaptation reference values for Fig. 8. Results use a two-hour window and a 30-min checkpoint; violations cover all training episodes, and – means 90% success was not reached.

Method	To 90% (min) ↓	Success @ 30 min (%) ↑	Violations (%) ↓
DSRL [47]	–	10	90
Residual off-policy [3]	35.1	80	20
Facet-0 local adaptation	10.5	95	5

F.1 Behavioral Analysis

This subsection records the mechanisms underlying the takeaways of Sec. 6. The body states the conclusion; the supporting detail is kept here.

Contact-aware retry. The largest margins over $\pi_{0.5}$ occur on RAM and CPU, where a visually aligned insertion can still meet the slot rim. The base policy continues the already issued downward chunk, whereas the adapted actor uses the resulting off-axis wrench to retreat, correct laterally, and re-enter (Fig. 8). The correction remains within the same sub-goal and runs at the contact-control rate, so it does not require the backbone to re-plan the task.

Early failure observability. During a press-and-lock sub-task, gripper occlusion can make a slight misseat look identical to successful seating. Wrist F/T separates the two cases: force rises while displacement stalls, allowing the system to flag the jam, re-issue the sub-goal, and complete the insertion. This contact-specific observability underlies the recovery gain in Sec. 6.3.

Protected recovery credit. Because Eq. (4) is evaluated per sub-task, a recovered trajectory receives credit for its corrective segment rather than being discarded with the preceding failure. Regime-conditioned selection then keeps these rare contact-recovery frames in the positive set. Together, the two mechanisms improve the tail behavior that determines whether a failed interaction resumes or halts the cell.

Separation of evaluation scales. Table 7 measures complete chains and individual contact decisions, Table 9 isolates changes from deployment post-training, and Table 10 records the speed and safety of local adaptation. These quantities retain separate denominators because they describe different operating scales; reading them together connects task completion to recovery and adaptation without collapsing distinct outcomes into a single aggregate score.