FORGE-plus: Force-Budgeted Recovery for Contact-Rich Assembly with a Frozen LLM Supervisor

A simulation study of who sets the force ceiling, and what to do on failure

Kyupaeck Jeff Rah
Independent Researcher
jeffrah89@gmail.com

Midum Oh
Independent Researcher
midumoh@gmail.com

July 2026

Abstract

Force-conditioned RL skills can seat tight-clearance assemblies while respecting a commanded force ceiling, but two questions remain open: *who sets the ceiling*, and *what should the robot do when an insertion fails*. Autotuning the ceiling for success works only for indestructible parts, and the natural recovery — press harder — is precisely the action that destroys a fragile one. We study a two-layer system in which a **frozen, text-only LLM** (a) sets a per-object force ceiling $F_{\max}$ from the object's identity before the episode, and (b) on failure, selects a recovery maneuver from a fixed menu by reading a **compact text force signature** — no images anywhere in the system. Force authority never belongs to the LLM: a hard clamp in the fast control loop saturates commanded forces to $F_{\max}$, the recovery menu contains no "increase the ceiling" option, and the hidden per-instance breaking force F_{break} is visible only to the evaluator, making the fragility metric non-circular. On a fragile bottle-into-rack placement and a 0.4 mm-diametral-clearance gear insertion, each executed on two grippers (Robotiq 2F-140 and Franka Panda hand), a single unified checkpoint passes strict 256/256 clean gates on both a fragile and a robust object class with zero breaks, decides its own release via a learned head (256/256, zero bad releases), and completes a fully physical table-pick flow at a 5.4 N mean peak force. Under an injected in-grip slip, the force-signature chain recovers 40% (2F-140) and 64% (Franka) of jams, while a press-harder baseline shows two distinct failure faces: it is 100% futile on one gripper and breaks 96% of fragile parts on the other. We report our negative results as first-class findings: PPO provably fails at this clearance on the 2F-140 (exploration noise that can find the funnel already breaks the part), tiny-std PPO polish destroys a working policy, and three natural designs for a learned release head fail for instructive reasons. All experiments are in rigid-body simulation with breakage modeled as a hidden scalar force threshold; no sim-to-real claim is made. Videos, code, and evaluation logs: https://robot-team00.github.io/FORGE-plus/.

1 Introduction

The Factory→IndustReal→AutoMate→FORGE line [1–4] has made simulated contact-rich insertion robust and, in several cases, transferable to hardware. FORGE [4] in particular conditions its policy on a maximum allowable force, so the skill can trade speed against gentleness. But that progress leaves two questions unanswered.

Who sets the ceiling? FORGE autotunes $F_{\max}$ (or takes it from a human) to maximize success. That is the right choice for indestructible parts and the wrong one the moment a single line

mixes a steel bolt with a nylon snap clip: the ceiling that seats the bolt strips the clip. The budget must be *per object*, and it has to come from somewhere — most naturally from what the object *is*.

What do you do on failure? Existing LLM/VLM failure reasoners — REFLECT, DoReMi, AHA [5–7] — read vision and re-plan at the task level. But the information that distinguishes a wedge from a cross-thread from a burr lives in the *force trace*, not the pixels: those failures often look identical on camera and feel completely different. And the obvious recovery — press harder, a published behavior in Tactile-VLA [8] — is exactly the move that finishes a robust part and destroys a fragile one.

FORGE-plus closes both gaps at once with a deliberately thin semantic layer:

- A **frozen LLM** reads the object's identity (text only, no images) and sets a per-object $F_{\max}$ before the episode starts.
- On failure, the same LLM reads a **compact force/contact signature** and picks a recovery from a fixed menu that has no "increase the ceiling" option.
- **Safety is enforced by a hard clamp in the fast control loop** — never by the language model. The LLM emits a number and a menu choice; the clamp decides what force is physically commanded.

Three invariants make the evaluation non-circular (§3.2): the hidden breaking force F_{break} is never observed by any learned or LLM component; $F_{\max}$ is immutable during recovery; and force authority lives in the fast loop. The single way to score well on breakage is to set and respect a sensible ceiling from what the part *is*.

Scope. This is a **simulation-only** study of the terminal "micro" phase of manipulation — the last few centimeters of contact-rich seating or insertion, where force decides success and breakage. No component uses vision: the skill observes proprioception, pose, and force/torque; the LLM reads text. Object identity is operator-provided and the poses of part and target are taken from privileged simulator state; perception, part identification, and long-range transport are upstream problems, deliberately out of scope. Breakage is modeled as a hidden scalar threshold on peak contact force — there is no fracture or deformation modeling — and no sim-to-real transfer is claimed.

Contributions.

1. **A two-layer architecture and benchmark** for force-budgeted assembly under per-object fragility, with a non-circular breakage metric (hidden per-episode F_{break} draws), seven evaluation metrics, and six baselines including a press-harder recovery modeled on Tactile-VLA's published behavior (§3, §4).
2. **Full-cycle results on two grippers.** On the Robotiq 2F-140, one unified checkpoint passes strict clean gates of 256/256 on both a fragile ABS gear and a steel gear with zero breaks, a learned release head passes 256/256 with zero bad releases, and a fully physical table-pick flow completes 64/64 at a 5.4 N mean peak insertion force — the gentlest contact of the project (§6.2).
3. **Force-signature recovery vs. baselines.** On the bottle task, a rim wedge is caught from the force signature at 16.1 N against a 23 N break, and budget-preserving retries seat the fragile bottle in 9/9 headless recovery episodes with zero breaks (§6.1). On the gear task, under a 5 mm in-grip slip, only the force-signature chain ever routes to `regrasp` — the one

maneuver that fixes a tilted grip — recovering 40% (2F-140) / 64% (Franka) of jams vs. 28% / 32% for a menu-sampling proxy of a vision-based reasoner and 0% for a hand-coded heuristic. Press-harder shows *two different failure faces* on the two grippers: futile (0% success, 0 breaks, 100% timeouts) on the 2F-140, destructive (96% breaks) on the Franka (§6.4).

4. **Negative results as findings.** PPO cannot learn this insertion on the 2F-140 at 0.4 mm clearance — the exploration noise needed to search the funnel already breaks the part (nine escalating runs, zero seats) — and tiny-std PPO polish destroys a working BC policy within 25 iterations. Three natural designs for a learned release head fail for reasons that generalize (§5). The oracle budget $F_{\text{break}}-\varepsilon$ breaks half the fragile parts because command clamping does not bound contact overshoot (§6.3).

2 Related work

The skill: FORGE. We reuse FORGE-style force conditioning [4] essentially wholesale: the policy takes a normalized force command as an observation, is penalized for overshooting it, and learns to seat under the ceiling. The difference is upstream — FORGE chooses $F_{\max}$ to make the task succeed; we choose it to keep the part alive, from identity, and the recovery layer is forbidden from relaxing it.

The closest semantics: Tactile-VLA. Tactile-VLA [8] performs language-conditioned force control and demonstrates "reasoning to overcome failure" by autonomously *increasing* force. That behavior is precisely our press-harder baseline. We differ in using a frozen LLM rather than a fine-tuned VLA, a force-only signature with no vision, a hard per-object ceiling recovery cannot cross, and an evaluation containing fragile parts that press-harder destroys. In Tactile-VLA's setting there is no breakage to violate, so force escalation is safe by construction; ours is the setting where it is not.

The closest safety architecture: PaCo-VLA. PaCo-VLA [9] shares the commitment that the network proposes while a runtime shield retains authority; its shield is a passivity/energy-tank contract over admittance updates, explicitly not a bound on peak force. Ours is exactly a hard scalar peak-force clamp tied to object identity. ForceVLA [10] and CompliantVLA-adaptor [11] provide further force-aware VLA context.

Failure reasoners: REFLECT, DoReMi, AHA. All three [5–7] are vision-centric and operate at the task-plan level (wrong object, missing step, mis-grasp). Our failure signal is a compact force/contact signature, the recovery is constrained by a force budget, and the failures we target live in contact dynamics that look alike on camera.

Assembly benchmarks. Factory [1], IndustReal [2], and AutoMate [3] established the simulation line we build on (we use Isaac Lab [12, 13]); none models per-object fragility or force-grounded recovery. Grasping embodiments follow GraspGen [14], whose two parallel-jaw grippers (Franka Panda hand, Robotiq 2F-140) are our generalization axis.

No single ingredient here is new; the contribution is the conjunction — force-grounded (not vision) semantic recovery, under a hard per-object ceiling the recovery cannot raise, scored closed-loop with a fragility metric the agent cannot read off.

3 System

Two layers run at different rates and never blur their roles (Figure 1). The **slow layer** is a frozen LLM called at most twice per episode — once at episode start to set the budget, once per failure to

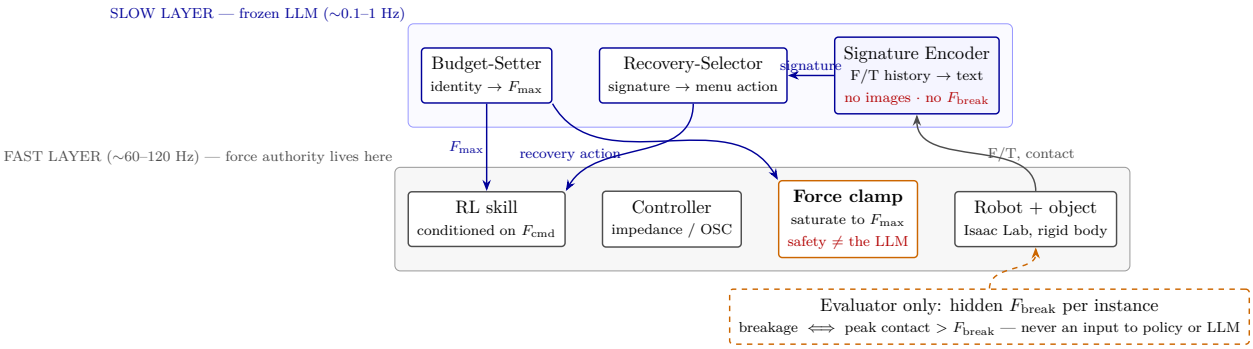

Figure 1: The two-layer architecture. The LLM only emits a number ($F_{\max}$) and a menu choice; the guarantee that commanded force never exceeds $F_{\max}$ is produced by the clamp in the fast loop, backed by a fixed 120 N global hard cap and JSON range validation. The hidden F_{break} feeds only the evaluator's breakage check.

pick a recovery — with JSON in and JSON out, no images. The **fast layer** is the force-conditioned RL skill plus an operational-space impedance controller at control rate, where force authority lives. A *signature encoder* between them turns the recent force/torque and contact history into a short text token.

3.1 The two LLM calls

Budget-setter. Input: the object's identity — name, material, class, nominal mass, geometry tags (e.g. `thin_wall`, `press_fit`) — plus the task and the global hard cap. Output: a numeric `F_max_N` (optionally per-axis), a confidence, and a rationale. The output is range-validated and clamped to a fixed global hard cap (120 N) before it reaches the controller; budget-setter outputs are cached per object class since the call is near-deterministic. The LLM reasons about the *class* — it can never memorize an instance, because F_{break} is sampled per episode (§4).

Recovery-selector. Input: the current $F_{\max}$ (echoed read-only), the attempt count, and the force signature:

```
{peak_axial_N, net_insert_mm, axial_rising, lateral_bias, contact_persist_ms,
                                                    slip_events}
```

Output: one action from the fixed menu `{retract_and_reapproach, wiggle_search, rotate_align, regrasp, abort}` with bounded parameters. The menu deliberately omits any "increase the ceiling" option, and the returned `keep_F_max_N` is overwritten to the original value server-side regardless of what the model returns. Recovery reallocates *motion*, not force. The recovery *primitives* are fixed scripted maneuvers; the LLM's contribution is the *selection*, grounded in the signature.

The LLM backend is swappable (hosted API, a local 7–8B instruct model, or a deterministic force-reasoned heuristic for CI); all backends read only text.

3.2 Design invariants (the non-circularity of the evaluation)

1. F_{break} **is never observed by the agent.** It is sampled once per episode from the object class's distribution, stored in the simulator, and read only by the evaluator's breakage check. The policy, the signature encoder, the budget-setter, and the recovery-selector never see it; a runtime assertion in `SignatureEncoder.encode()` guards this at test time.

2. $F_{\max}$ **is immutable during recovery.** The recovery-selector's `keep_F_max_N` is always overwritten back to the original value before it reaches the controller.

3. **Force authority is in the fast loop.** The `ForceClamp` saturates the commanded wrench before every simulation step regardless of what the LLM said; a fixed 120 N global cap is a second line of defense against hallucinated ceilings.

One subtlety we do not hide: clamping the *command* does not by itself bound the *contact* force — impedance overshoot can exceed $F_{\max}$. We mitigate with low controller stiffness and bounded per-step motion, and we measure the gap as a first-class metric (clamp fidelity). Section 6.3 shows this is not a technicality: it is why the oracle budget breaks parts.

3.3 The closed loop

Each episode runs the skill until success or a detected failure; on failure the signature is encoded, the LLM picks a recovery, the fixed primitive executes within the same $F_{\max}$, and the skill resumes — up to $K_{\max}$ attempts. Failure detection is deliberately conservative and force-grounded: a *jam* is sustained contact with no net descent over a window (a wedge, not a force spike — a clean insertion grind that does descend passes through), and a *contactless hover* branch catches the policy correctly refusing a geometrically impossible insertion (§6.4). A false positive costs one benign lift-and-retry attempt, never more force.

4 Benchmark

4.1 Tasks, objects, embodiments

Bottle placement — a fragile bottle into a wine rack. The robot inserts a LIBERO wine bottle, held by the neck under real friction, into a cell of a 3×3 rack, then releases it standing upright — a place-and-release peg-in-hole where "press harder" is maximally destructive. Fragile class: $F_{\text{break}} \approx 22$ N, budget 8.8 N; robust class: 180 N. (The rendered mesh is the wine bottle in all cases; the class sets the hidden fragility.) The recovery episode is demonstrated on both grippers (Robotiq 2F-140 and Franka Panda hand); the quantitative arc is on the Franka.

Gear insertion — tight clearance. The FORGE GearMesh medium gear (∅35.5 mm hub, bore fit band $r = 5.15$ mm, **0.4 mm diametral clearance**) must be inserted onto the middle shaft of a three-shaft base plate. Two object classes share the geometry: `abs_gear` (fragile; F_{break} drawn per episode from 38 ± 5 N; identity-derived budget ≈10 N insertion) and `steel_gear` (robust; 100 N budget). Success is a *strict true seat*: gear origin within 0.6 mm of the seated height, upright, on the correct shaft. The full quantitative arc runs on **both grippers**: the Robotiq 2F-140 four-bar adaptive gripper and the Franka Panda hand.[1]

Disturbances. Bottle placement injects a lateral base-aim error that wedges the bottle on the cell rim. Gear insertion injects a **5 mm in-grip slip**: the gear is displaced in the pinch mid-carry. The pads re-center the *position* within a substep, but the *orientation* does not recover — the asymmetric squeeze leaves the gear tilted 7–10° in the grip, which cannot thread a 0.4 mm-clearance bore (∼3–4° max while threaded).

[1]The 2F-140 port required building a combined Franka+2F-140 USD asset (NVIDIA ships that combination only for the 2F-85) and two PhysX-level findings documented in the repo: the four-bar loop joints do not survive articulation teleports (hence a no-teleport contract: spawn at the authored pose, drive only by position targets), and the merged gripper's excluded loop joints materialize only when a standalone instance of the same gripper USD exists in the scene.

4.2 What is learned vs. what is scripted

The split is explicit, and the released videos label every phase on-screen in real time (green LEARNED / orange SCRIPTED):

Learned: the force-guided insertion — funnel search, alignment, descent, seating press, everything from the entrance hand-off to the seat — and the *release decision* (an 8th action dimension; the environment opens the gripper only when the learned head crosses its threshold).

Scripted: pre-approach transport between known poses (episode staging, as in the underlying benchmarks), the recovery *primitives* (the menu the LLM selects from), and the post-release hand retract. In the final table-pick flow, nothing about the object is teleported, pinned, or kinematically attached at any point: the pick closes a real friction grip on the resting gear, and everything after is genuine physics.

4.3 Metrics and baselines

Seven metrics: (1) closed-loop multi-attempt success; (2) **breakage rate** (peak contact $> F_{\text{break}}$; non-circular since F_{break} is evaluator-only); (3) budget appropriateness (the margin $m = F_{\text{break}} - F_{\max}$, over-/under-budget rates); (4) recovery efficacy; (5) force economy (peak/mean contact per success); (6) clamp fidelity (contact overshoot above $F_{\max}$); (7) cross-gripper transfer.

Six baselines: *no-ceiling*, *fixed global ceiling*, *press-harder* (per-object budget, recovery escalates force $\times 1.25$ per attempt — Tactile-VLA's published recovery behavior), *heuristic recovery* (hand-coded rules on the same signature), *vision-LLM proxy* (a menu-sampling stand-in for pixels-only reasoners — see the caveat in §6.4), and the *oracle ceiling* ($F_{\text{break}} - \varepsilon$; cheats by reading the hidden variable).

5 Skill learning: what works at 0.4 mm, and what provably does not

The skill is a force-conditioned MLP policy (34-dim observation: proprioception, EE pose, F/T, object-up axis, phase; F_{cmd} conditions the network) whose position deltas ride on an operational-space impedance controller with bounded per-step motion. On the Franka hand, FORGE-style PPO [15] training suffices (clean gate: 200/200 strict seats, 0 breaks). Porting to the Robotiq 2F-140 exposed three plant-level defects that made every policy fail before learning was in question — the four-bar's paired knuckles must be driven symmetrically, the wrist target must be anchored and command-continuous at hand-off, and the funnel search needs lateral authority while in contact — and then a clean negative result:

5.1 PPO provably fails at this clearance (negative result 1)

The funnel tolerance ($\sim$1.5 mm capture radius; 0.25 mm effective bore clearance at $\mu = 1.0$) sits far below workable exploration noise: an action std of 0.12 already breaks 40/64 fragile gears while seating 1/320. **Nine escalating PPO runs on the 2F-140 — spanning plant fixes, noise schedules, annealing, and contact-observation variants — never seated a single gear**; all nine checkpoints are kept in the repository manifest as evidence. The exploration noise needed to *search* the funnel is already destructive at the funnel's own scale: for fragile tight-clearance assembly, the exploration–damage trade-off can exclude on-policy RL outright.

A second, sharper negative: **tiny-std PPO polish destroys a working policy**. Fine-tuning a functioning BC policy with PPO at std 0.05 collapsed it from 84% to 0/32 within 25 iterations, at

every snapshot. The mechanism is the $1/\sigma^2$ term in the Gaussian policy gradient: as std shrinks, likelihood-ratio gradients are amplified so that the mean moves far more coarsely than the 0.4 mm task tolerates.

5.2 What works: demos → DAgger → BC → weight soup

Scripted experts are permitted for *training data only* (never in evaluation). The pipeline: collect expert demonstrations, run DAgger [16] rounds relabeling the policy's own visited states, and fit the deterministic mean by behavior cloning. The lineage climbs from closed-loop failure (plain BC: 30/32 broke) through DAgger rounds to `bc3` (fragile class 256/256) and `bc7` (best single: steel 32/32, fragile 84%). The two classes tug the policy in opposite directions — the fragile class sits on its own break floor (F_{break} draws as low as ∼20 N vs. 15–25 N contact transients) — and balanced-refit attempts see-sawed.

Unification is a weight soup [17]: `uni` = 0.15·`bc3`+0.85·`bc7`. The interpolation is valid because `bc7` was warm-started from `bc3` (same basin); every $\alpha \in [0.10, 0.75]$ passes triage, steel peak force is monotone in α, and $\alpha = 0.15$ minimizes the joint force tails. One checkpoint then passes both clean gates (§6.2).

5.3 The learned release head (negative results 2–4, and the winner)

The policy should decide *when to let go* — an 8th action dimension — without disturbing the proven arm behavior. Three natural designs fail for general reasons:

1. *Frozen-trunk head trained on all timesteps* learns "open iff already open" and never fires at deployment. The environment latches the first release, so every post-open observation carries no decision information — **post-open samples are label poison**; drop them always.

2. *Pre-open-only labels on the frozen trunk* yields 27% false positives / 52% false negatives: the arm-trained trunk provably *discards* the seat-state signal, having never needed it.

3. *Trunk fine-tuning with an arm-distillation loss* sees the two losses fight; the false-negative rate pins at 50% with zero usable coverage.

The winner is an input-skip head: a single linear layer reading the *raw observation* (plus F_{cmd}), bypassing the trunk. A logistic-regression diagnostic first proved the release signal is linearly separable in the raw observation (zero false negatives, zero-false-positive window coverage 255/256), so only that linear layer is trained — the arm dimensions and trunk stay bit-identical to `uni`. Two deployment traps with teeth: the feature standardization must be folded into the weights and the operating threshold set *after* folding (fp32 fold error scales with logit magnitude); and the collection-time settle matters — labeling release 20 steps after the geometric seat let the seating press continue and *broke 36% of fragile gears before the release could fire*; a 5-step settle gives 256/256. The head reads a phase one-hot, so it is staging-specific and must be recollected per deployment flow.

5.4 Physical staging: the table pick and the place-on-table regrasp

The last non-physical assist in the pipeline was a seat-window pin that placed the gear in the closed gripper at episode start. The final flow replaces it end-to-end: the gear spawns resting on the table, the gripper descends open, closes a real friction grip on the hub, lifts, carries, and hands off to the policy. Two durable lessons: precise transports must run on stiff joint drives, not the

Figure 2: The rendered fragile recovery episode (Franka, glass-class bottle, $F_{\text{break}} \approx 23\,\text{N}$). The JAM card shows the actual signature the LLM saw — peak 16.1 N, net insert 1.6 mm, rising, lateral bias `+x steady` — its `rotate_align` decision, and the standing constraints: *from the force signature alone (no vision)*; *same* $F_{\max}$ *— never press harder.*

operational-space controller — an OSC wrist-twist disturbance walked the end-effector 15–30 mm sideways during any settle or traverse, and this same defect independently killed the carry, the hover, and the recovery place-traverse before being rooted out — and grip geometry tolerances are load-bearing (pads 6 mm low on the hub produce a 3.5° carry lean, which the policy correctly *refuses* to insert). In recovery, `regrasp` becomes fully physical: place the tilted gear back on the table, open, re-pick it upright, carry back, resume. Safety rules found by root-causing (each with a recorded incident): drives may engage only in confirmed free space (<0.5 N contact; the timeout-to-drives path once crushed a wedged gear at 213 N), every drive engage must droop-compensate (∼3.5 cm sag under load), and the place descent must gate on the realized gear height, not the arm pose (a held gear rides ∼30 mm lower than a resting one; 319 N observed otherwise).

6 Experiments

All results are in Isaac Lab rigid-body simulation with the deterministic policy mean, strict success criteria, and hidden per-episode F_{break} draws. Every number below is reproducible from the repository's evaluation scripts and is stated with its protocol.

6.1 Bottle placement: a jam caught at 16.1 N against a 23 N break

The recovery episode is rendered on both grippers. On the Robotiq 2F-140, a seeded base-aim fault wedges the fragile bottle on the rack; the jam is caught from the force signature (peak 13.9 N against that episode's 19 N break), and the second attempt places the bottle upright. The quantitative arc is on the Franka: a FORGE-style PPO policy (8-dim action including the learned release) performs the insertion under gentle contact (∼0–4 N against the fragile budget) at ≈68% single-attempt success; the recovery loop is what makes it reliable. With an induced rim wedge on the *fragile* class, the jam is caught from the force signature at **16.1 N — far below the episode's 23.3 N break** — the LLM picks `rotate_align`, and the learned policy re-descends and seats the bottle: **9/9 headless recovery episodes seat the fragile bottle** (2–3 attempts each, 0 breaks), even though a single attempt under the low budget rarely seats it. The same loop on the robust class catches the wedge at ∼17 N vs. a 180 N break. The honest value of the layer is exactly this: the conservative detector plus benign, budget-preserving retries turn an imperfect learned insertion into a reliable

Table 1: Robotiq 2F-140 clean gates (deterministic policy, strict true-seat criterion, hidden per-episode F_{break}; 256 episodes per class unless noted). Release success is stricter than insertion success: released + gear standing seated + hand retracted clear, with the environment opening the gripper only on the learned head's own threshold crossing.

Milestone	Checkpoint	Fragile ABS gear	Steel gear
Clean insertion (unified)	`uni`	**256/256, 0 breaks**; peak 15.9 N mean / 18.5 p95	**256/256, 0 breaks**; 35.6 / 57.6
+ learned release	`uni_rel`	**256/256, 0 breaks, 0 bad releases**; 15.7 / 18.5 / 20.1 max	**256/256, 0, 0**; 35.7 / 59.3 / 69.6
Full table-pick flow	`uni_rel_tp`	**64/64 (smoke), 0 breaks, 0 bad rel.**; peak **5.4 N mean** / 5.8 max	—

Figure 3: Stills from the released 2F-140 videos (HUD labels every phase LEARNED/SCRIPTED with a live force gauge vs. F_{max} and F_{break}). Left to right: the scripted-staging table pick (a real friction grasp of the resting gear); the learned release firing at 1.9 N after a clean insertion; a recovery episode mid-`regrasp` — the tilted gear has been placed back on the table for a physical re-pick (LLM decision `wedge` → `regrasp`, attempt 2); the same episode ending seated, released, hand clear after two full regrasp cycles.

one, whether the stall came from the injected misalignment or the policy's own marginality. (The rendered video's stricter full-place gate — upright settle after a learned release the head was never trained to time on this object class — passes in about 1 of 3 takes; the release dimension is gated off during recovery and sampled at the seat, as documented in the repo.)

6.2 Clean gates: one checkpoint, both classes, zero breaks

Table 1 summarizes the Robotiq 2F-140 arc; the Franka clean gate is 200/200 strict seats, 0 breaks (peak 13.8 N mean / 15.9 p95 / 17.6 max). The unified checkpoint records zero over-budget episodes in both classes.

Two observations. First, the weight-soup unification is not cosmetic: no single BC checkpoint passed both class gates, and PPO could not train one (§5.1). Second, an instructive inversion: the fully physical *table-pick* flow produces far *gentler* insertions (5.4 N mean peak) than the pinned staging it replaced (15.9 N). The staging pin left a small systematic grip offset that the policy paid for during the funnel search; a real friction pick centers the hub in the pads better than the pin ever did. Removing the last non-physical assist made the force economy better, not worse. (The 64-episode table-flow gate is a smoke, not a full 256-episode gate; the steel-class table-pick smoke is open work.)

6.3 Budget baselines: the oracle breaks half the parts

Table 2 gives a stronger separation than we expected — **ours** $\gg$ **oracle** $>$ **fixed** — with one inversion that is a finding in itself. The oracle sets $F_{\text{max}} = F_{\text{break}} - \varepsilon$, the "optimal" budget if clamp fidelity were perfect. It is not: conditioned at 33 N, the skill's funnel-entry overshoot peaks at 45–54 N ($\sim$1.5$\times$ the budget), past the 38 ± 5 N breaking force in half the episodes. **Budget appropriateness must cover the overshoot distribution, not just sit under** F_{break} — the conservative identity-derived 10 N budget absorbs the same overshoot with room to spare, seating 100% with zero breaks. The fixed 60 N global ceiling destroys every part (the canonical argument for per-object budgets), and the unbounded baseline is not even unsafe — it is non-functional,

Table 2: Budget-setter baselines on the fragile ABS gear (Franka, deterministic mean, ∼200 episodes/cell, strict criterion). Margin = mean(F_{break}) − $F_{\max}$. *No-ceiling produced *zero episode completions*: F_{cmd} conditioning far outside the training distribution degenerates behavior (no descent to contact) — non-functional rather than safe.

Budget	$F_{\max}$	Success	Breakage	Peak force mean/p95/max (N)	Over-budget eps
Ours (LLM, identity-only)	10 N	**1.000**	**0.000**	14.9 / 16.8 / 20.2	0/200
Oracle ($F_{\text{break}} - 5$)	∼33 N	0.502	0.498	33.9 / 44.9 / 53.8	0/201
Fixed global	60 N	0.000	1.000	34.6 / 42.2 / 47.1	205/205
No ceiling*	120 N	0.000	0.000	—	200/200

Table 3: Jam-recovery sweep: fragile ABS gear, 5 mm in-grip slip every episode, 25 episodes/cell, deterministic mean, identity-derived budget, $F_{\max}$ never raised. *The two tables use different step caps (1600 vs. 700, because one 2F-140 regrasp cycle is ∼450 steps and the cap must fit ∼3 cycles) and different checkpoints, so rows are comparable within a table, not across tables.* On the Franka table the attempts counter saturated at 5 in every cell (cosmetic settle-churn re-fires) and is omitted; the 2F-140 protocol suppresses that churn.

Robotiq 2F-140 (1600-step cap; unified checkpoint, clean gates 256/256 both classes)					
Recovery	Success	Breaks	Timeouts	Attempts	Peak force mean/max (N)
Ours (force signature)	**40%** (10/25)	12% (3)	48% (12)	3.56	23.0 / 48.3
Vision-LLM proxy	28% (7/25)	12% (3)	60%	3.88	19.0 / 43.6
Heuristic	0%	0%	100%	5.00	12.5 / 18.0
Press-harder	0%	0%	100%	5.00	11.3 / 16.5
None	0%	**20% (5)**	80%	0	10.2 / 50.5
Franka Panda hand (700-step cap; clean-gate policy 200/200, 0 breaks)					
Recovery	Success	Breaks	Timeouts		Peak force mean/max (N)
Ours (force signature)	**64%** (16/25)	12% (3)	24% (6)		23.2 / 31.5
Vision-LLM proxy	32% (8/25)	0%	68%		15.0 / 31.6
Heuristic	0%	0%	100%		0.0 / 0.0
Press-harder	4% (1/25)	**96% (24)**	0%		36.1 / 44.8
None	0%	0%	100%		0.0 / 0.0

because force conditioning is part of the skill's operating envelope and a far-out-of-distribution F_{cmd} degenerates the behavior itself.

6.4 Recovery under an in-grip slip: two grippers, five baselines

The failure mechanism. The 5 mm slip leaves the gear *tilted 7–10° in the grip* (position recovers within a substep; orientation never does — probe: 4.6° →10.1° at the kick, no decay). A tilted bore cannot thread the 0.4 mm fit band, so the deterministic policy backs off and *hovers at zero contact force* — correctly refusing a geometrically impossible insertion, and invisible to any force-threshold jam detector. A contactless-hover branch in the failure detector makes the refusal legible; the signature chain routes a *recurring* hover (an anomaly that survives an arm maneuver, hence travels with the part) to `regrasp`, the only menu action that fixes an in-grip tilt. The re-seat restores the canonical grip (9.8° →0.2°).

Reading Table 3. Only the force-signature chain ever restores an insertable grip. The hand-coded heuristic never reads the hover signature, maps every zero-force state to `wiggle_search`, never regrasps: 0%. The vision-LLM proxy — *implemented as a random menu draw*, a deliberately

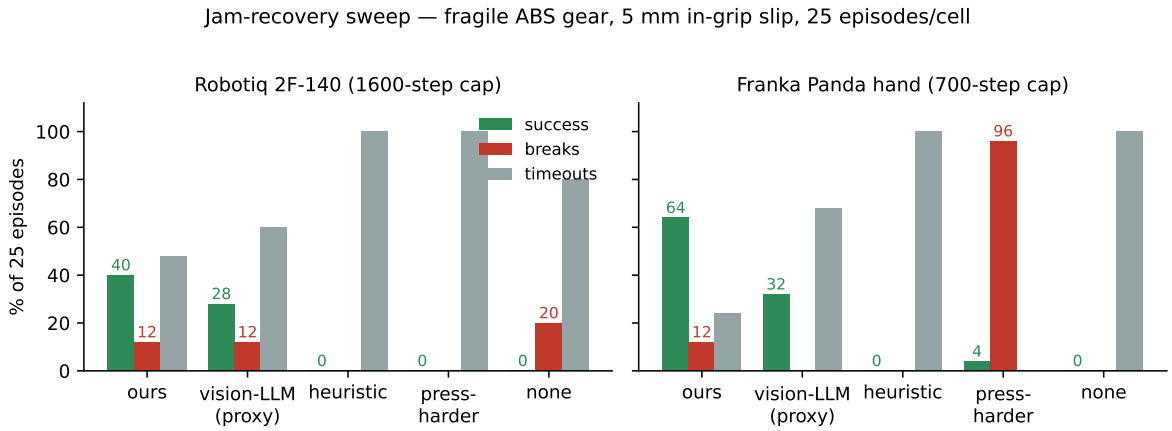

Figure 4: The recovery sweep of Table 3 as grouped bars. Note press-harder's two faces: futile on the 2F-140 (100% timeouts, zero breaks — the escalated ceiling never engages because the tilted bore never takes load; there, doing *nothing* is the destructive cell, 20% breaks), destructive on the Franka (96% breaks).

generous stand-in for pixels-only reasoners that cannot see in-grip force asymmetry — succeeds exactly as often as it happens to pick `regrasp`: the force signature is worth 40% vs. 28% on the 2F-140 and 64% vs. 32% on the Franka (the gap is narrower on the 2F-140 because a wasted wrong-maneuver attempt burns a ∼450-step regrasp cycle from the step budget). We label this baseline a proxy: it bounds what any reasoner that cannot discriminate the signature could achieve by selection alone.

Press-harder shows two faces of the same anti-pattern. On the 2F-140 it is *futile*: its only maneuver never fixes the in-grip tilt, the tilted bore never takes load, so the raised ceiling never even engages — 0 breaks, 0 successes, 100% timeouts at a 16.5 N max. On the Franka it is destructive: escalating $F_{\max}$ ×1.25 per attempt destroys **24 of 25** fragile gears (peaks to 44.8 N vs. F_{break} 38 ± 5). Both faces refute force escalation as a recovery strategy, for different reasons; on the 2F-140 the destructive face reappears in the *none* cell, where with no recovery interrupting it the policy's own descent eventually presses the tilted bore into the shaft tip (5/25 breaks at up to 50.5 N). The jam is real, and doing nothing is worse than a wrong maneuver.

Robust class. On the steel gear the same chain recovers **25/25 with 0 breaks and 0 timeouts** — but recovery episodes press 85 N mean / 135 N max vs. the 35.6/62.5 clean gate, consuming most of steel's 100 N budget margin.

Honest caveats. (i) Recovery episodes press harder than clean ones on both grippers (2F-140: 23.0 N mean / 48.3 max vs. 15.9 / 36.7 clean) — that envelope exposure, not any raised ceiling, is what catches low F_{break} draws and accounts for all 3 fragile breaks in each *ours* cell; a gentler post-recovery descent profile is the obvious next refinement. (ii) The 48% timeout rate in the 2F-140 *ours* cell is the current frontier, not breakage: the cap fits ∼3 recovery cycles and detector churn can burn them. (iii) With the fully physical place-on-table regrasp and learned release enabled end-to-end, a 5-episode smoke completes 3/5 (2 breaks, 0 timeouts) — a smoke, not a cell; the 25-episode table-flow cell is open work.

6.5 Force economy across the arc

Peak insertion forces on the fragile class, all with the identity-derived ≈10 N budget and zero raised ceilings: pinned-staging clean gate 15.9 N mean; with learned release 15.7 N; full table-pick flow **5.4 N mean / 5.8 max**; recovery episodes 23.0 N mean / 48.3 max (envelope exposure, §6.4).

The system's gentlest contact came from *removing* scaffolding, and its hardest pressing comes from post-recovery seating — the measured cost of a second chance.

7 Limitations

Simulation only; breakage is a scalar. No real robot, no sim-to-real claim. Breakage is a hidden threshold on peak contact force in rigid-body physics — no fracture, deformation, or fatigue. FORGE-style force conditioning is independently known to transfer [4]; hardware is deferred, not implied.

The clamp bounds the command, not the contact. Impedance overshoot reaches $\sim$1.5$\times$ the budget at funnel entry (§6.3). We surface it as the clamp-fidelity metric and absorb it with conservative budgets; a contact-force kill switch is an open design choice.

Disturbances are injected and simple. The rim wedge and the 5 mm slip are controllable, reproducible faults — not a naturalistic failure distribution. Richer jams (angular catch, burrs, cross-threading) are future work.

The vision baseline is a proxy. We compare against a menu-sampling stand-in, not a full REFLECT/AHA pipeline with rendered frames; the comparison bounds selection-only performance rather than testing a specific vision system.

Scripted components remain, and are labeled. Transport staging and recovery primitives are scripted (the LLM selects; the primitive executes); the release head is staging-specific (it reads a phase one-hot) and must be recollected per flow. Several table-flow results are explicitly smokes (64-episode release gate; 3/5 regrasp cell), and the 2F-140 recovery cell's 48% timeout rate is an open frontier.

LLM scope. The budget-setter reasons over provided identity text; we did not evaluate adversarial or paraphrased identities, and budget outputs are cached per class. Whether the identity-derived budget is fully gripper-invariant is measured only indirectly here (the same class budgets drive both grippers' evaluations).

8 Conclusion

FORGE-plus asked two questions that force-conditioned assembly skills leave open: who sets the ceiling, and what to do on failure. The answers that survive our evaluation are structural, not model-flavored. A conservative, identity-derived budget beats the oracle, because a real budget must cover the controller's overshoot distribution — not flirt with F_{break}. A recovery layer earns its keep only if it can *discriminate* failures: the force signature is worth a factor of $\sim$2 over selection-by-luck on both grippers, entirely because it is the only signal that routes a recurring contactless hover to the one maneuver that fixes a tilted grip. And "press harder," the natural recovery, fails in two distinct ways on two grippers — destructive where the load path engages, futile where it does not — while never once being the right answer. Meanwhile, the safety story never depended on the language model: the clamp did.

We are equally interested in what did not work. On-policy RL is excluded at 0.4 mm clearance by the exploration–damage trade-off itself; the practical pipeline is expert demos, DAgger, BC, and a weight soup; a learned release decision needs an input-skip head because trained trunks provably discard the state it depends on. We release the code, the evaluation scripts, the checkpoint manifest (including the nine failed PPO lineages kept as evidence), and the labeled videos.

References

[1] Yashraj Narang, Kier Storey, Iretiayo Akinola, Miles Macklin, Philipp Reist, Lukasz Wawrzyniak, Yunrong Guo, Adam Moravanszky, Gavriel State, Michelle Lu, Ankur Handa, and Dieter Fox. Factory: Fast contact for robotic assembly. In *Robotics: Science and Systems (RSS)*, 2022. arXiv:2205.03532.

[2] Bingjie Tang, Michael A. Lin, Iretiayo Akinola, Ankur Handa, Gaurav S. Sukhatme, Fabio Ramos, Dieter Fox, and Yashraj Narang. IndustReal: Transferring contact-rich assembly tasks from simulation to reality. In *Robotics: Science and Systems (RSS)*, 2023. arXiv:2305.17110.

[3] Bingjie Tang, Iretiayo Akinola, Jie Xu, Bowen Wen, Ankur Handa, Karl Van Wyk, Dieter Fox, Gaurav S. Sukhatme, Fabio Ramos, and Yashraj Narang. AutoMate: Specialist and generalist assembly policies over diverse geometries. In *Robotics: Science and Systems (RSS)*, 2024. arXiv:2407.08028.

[4] Michael Noseworthy, Bingjie Tang, Bowen Wen, Ankur Handa, Chad Kessens, Nicholas Roy, Dieter Fox, Fabio Ramos, Yashraj Narang, and Iretiayo Akinola. FORGE: Force-guided exploration for robust contact-rich manipulation under uncertainty. *IEEE Robotics and Automation Letters*, 2025. arXiv:2408.04587.

[5] Zeyi Liu, Arpit Bahety, and Shuran Song. REFLECT: Summarizing robot experiences for failure explanation and correction. In *Conference on Robot Learning (CoRL)*, 2023. arXiv:2306.15724.

[6] Yanjiang Guo, Yen-Jen Wang, Lihan Zha, Zheyuan Jiang, and Jianyu Chen. DoReMi: Grounding language model by detecting and recovering from plan-execution misalignment. *arXiv preprint arXiv:2307.00329*, 2023.

[7] Jiafei Duan, Wilbert Pumacay, Nishanth Kumar, Yi Ru Wang, Shulin Tian, Wentao Yuan, Ranjay Krishna, Dieter Fox, Ajay Mandlekar, and Yijie Guo. AHA: A vision-language-model for detecting and reasoning over failures in robotic manipulation. *arXiv preprint arXiv:2410.00371*, 2024.

[8] 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.

[9] Yifan Cao et al. PaCo-VLA: Passivity-shielded compliance prior for contact-rich vision-language-action manipulation. *arXiv preprint arXiv:2606.00515*, 2026.

[10] Jiawen Yu et al. ForceVLA: Enhancing VLA models with a force-aware MoE for contact-rich manipulation. *arXiv preprint arXiv:2505.22159*, 2025.

[11] Wei Zhang, Ying Huang, et al. CompliantVLA-adaptor: VLM-guided variable impedance action for safe contact-rich manipulation. *arXiv preprint arXiv:2601.15541*, 2026.

[12] Mayank Mittal et al. Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning. *arXiv preprint arXiv:2511.04831*, 2025.

[13] Mayank Mittal, Calvin Yu, Qinxi Yu, Jingzhou Liu, Nikita Rudin, David Hoeller, Jia Lin Yuan, Ritvik Singh, Yunrong Guo, Hammad Mazhar, Ajay Mandlekar, Buck Babich, Gavriel State, Marco Hutter, and Animesh Garg. Orbit: A unified simulation framework for interactive robot learning environments. *IEEE Robotics and Automation Letters*, 2023. arXiv:2301.04195.

[14] Adithyavairavan Murali et al. GraspGen: A diffusion-based framework for 6-dof grasping. *arXiv preprint arXiv:2507.13097*, 2025.

[15] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[16] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011. arXiv:1011.0686.

[17] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning (ICML)*, 2022. arXiv:2203.05482.