CRAVE: RECOVERING PROGRESS FROM REPEATED DEMONSTRATIONS FOR CONTACT-RICH ROBOT POLICY POST-TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Progress-conditioned robot post-training can emphasize the brief transitions that determine contact-rich task completion, but its supervision typically requires manual semantic stages, a separately trained value estimator, or elapsed-time proxies that confuse time with task advancement. We introduce Cross-Episode Recurrent Abstraction via Multimodal Evidence (CRAVE), an offline relabeler that recovers an episode-consistent progress coordinate from visual–state configurations that recur across demonstrations. Unlike framewise temporal proxies, CRAVE couples assignments over each complete episode and converts future progress into ordinal training conditions for $\pi_{0.5}$ while leaving its objective and deployed architecture unchanged. Mechanistically, on 300 duration-balanced held-out garment-folding episodes, trajectory coupling localized human-referenced phase boundaries 12.72 s more accurately than framewise assignment (95% paired-bootstrap confidence interval, 11.38–14.09 s). Downstream, across 180 real-robot rollouts from nine matched policy checkpoints, Direct CRAVE completed 11/20, 15/20 and 16/20 trials at 150, 300 and 450 demonstrations, versus 10/20, 14/20 and 16/20 for the strongest matched comparator (plain behavior cloning or an adapted historical χ_0 Stage Advantage labeler). Across tasks, the same relabeling interface transferred to nail painting and low-data writing: observed mean four-stage completion was 35% for CRAVE versus 30% for supervised fine-tuning (SFT) over 20 nail-painting episodes per route, and the CRAVE-labeled writing policy completed 16/19 unique rollouts after training on 55 demonstrations. Operationally, a common eight-A100 engineering projection placed CRAVE label construction at 3.7–5.6 min versus 66–137 min for inference and export from an existing Stage Advantage checkpoint. Repeated demonstrations can therefore provide progress supervision for contact-rich VLA post-training without human phase labels, on-policy collection, or a separately trained reward, value or advantage estimator.

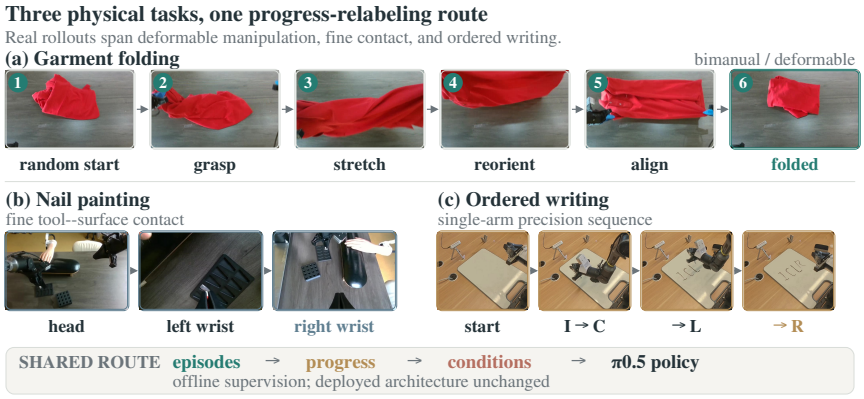

Figure 1: Real-robot scope of CRAVE: **(a)** bimanual garment folding from clutter, **(b)** multi-view fine-contact nail painting and **(c)** ordered single-arm writing. One offline progress-relabeling route supplies policy conditions across all three tasks without altering the deployed architecture.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Contact-rich manipulation turns on short physical transitions: establishing a grasp, preserving tool–surface contact, aligning deformable material, or recovering after a slip. Repeated demonstrations also contain long holds, repositioning, and operator-dependent timing. Plain behavior cloning (BC) fits these frames according to their frequency, without distinguishing how strongly each transition advances the task. Task-specific post-training thus needs supervision that identifies progress-bearing actions without discarding the broad control prior of a generalist policy.

Progress-conditioned post-training can expose these transitions to a vision–language–action (VLA) policy, but constructing its supervision remains costly. The Stage Advantage module in χ_0 predicts stage-conditioned advantages from pairwise image inputs and manually annotated semantic stages (Yu et al., 2026). RECAP instead learns a distributional value function from demonstrations, deployment experience, task outcomes and optional expert corrections before advantage-conditioned policy extraction (Physical Intelligence et al., 2025a). PLD acquires new policy-aligned trajectories with residual RL specialists and distils them into the generalist through supervised fine-tuning (Xiao et al., 2025). These routes add useful supervision or experience. We study the complementary fixed-corpus setting, where repeated demonstrations are already available and the goal is to construct progress conditions without manual stages, new deployment collection or a separate reward, value or advantage estimator. In this setting, normalized time advances through stalls, while UR-VC corrects its time-derived proxy through framewise cross-episode visual retrieval (Zhao et al., 2026). The remaining question is whether the corpus itself supports a task-relative progress coordinate that remains coherent over each complete episode.

We introduce CRAVE, a task-specific offline relabeler built on one observation: configurations that recur across independently collected episodes define a shared progress coordinate even when individual executions pause, skip a phase, or move backward. CRAVE first discovers recurrent prototypes in frozen visual and robot-state features. It then couples assignments over each complete episode to decode an anchored progress path, and ranks finite-horizon progress increments into ordinal text conditions. The resulting dataset trains a policy initialized from $\pi_{0.5}$ under the inherited flow-matching architecture and objective (Black et al., 2025; Physical Intelligence et al., 2025b). Direct CRAVE denotes this corpus-to-condition route: no parametric reward, value or advantage estimator sits between the demonstrations and the policy labels, and no CRAVE module runs at deployment.

CRAVE advances task-specific post-training in three ways. First, it introduces cross-episode recurrence as a source of progress supervision, allowing repeated demonstrations to reveal task structure without treating elapsed time as progress. Second, it turns that structure into an end-to-end offline relabeling route—from multimodal prototypes and trajectory-coupled progress recovery to ordinal policy conditions—while preserving the base VLA architecture and adding no learned CRAVE component at deployment. Third, it establishes a reusable fixed-corpus route across deformable manipulation, fine-contact tool use and precision writing, replacing manual stage annotation and a separately trained labeling model with direct structure recovery. Together, these contributions make progress-conditioned VLA post-training available from demonstration corpora alone.

2 RELATED WORK

Task adaptation and progress supervision. Robot policy learning spans task-specific behavior-cloning architectures (Mandlekar et al., 2022; Zhao et al., 2023; Chi et al., 2023) and generalist vision–language–action (VLA) policies trained across robots and tasks (Brohan et al., 2023; Kim et al., 2025; Octo Model Team et al., 2024). π_0 introduced a pretrained vision–language backbone with a flow-matching action expert, and $\pi_{0.5}$ extended this family through heterogeneous co-training and high-level semantic prediction (Black et al., 2025; Physical Intelligence et al., 2025b). CRAVE keeps this backbone unchanged and instead relabels the task corpus so that brief progress-bearing contact transitions are not outnumbered by holds and corrections during supervised fine-tuning.

Policy-improvement supervision can come from desired returns, learned critics, or newly collected behavior. Return-conditioned supervised learning and Decision Transformer place desired outcomes in the policy input (Emmons et al., 2022; Chen et al., 2021), whereas AWR, AWAC, IQL and critic-guided flow policies use learned values or advantages to shape policy fitting (Peng et al., 2019;

Nair et al., 2020; Kostrikov et al., 2022; Nguimatsia Tiofack et al., 2026). RECAP trains a distributional value function from demonstrations, deployments and outcome feedback (Physical Intelligence et al., 2025a); PLD collects policy-aligned recovery trajectories with residual RL specialists (Xiao et al., 2025). χ_0 learns stage-conditioned action quality from manually annotated semantic stages (Yu et al., 2026). Our matched comparison uses only its fixed historical Stage Advantage estimator as a label source, mapped to the same ordinal interface as CRAVE. CRAVE instead derives progress conditions from a fixed corpus without new behavior or a separately trained reward, value or advantage estimator.

Cross-episode progress recovery. Normalized time advances through stalls and regressions. UR-VC corrects this proxy frame by frame by retrieving visually similar states from other episodes and aggregating their time-derived labels (Zhao et al., 2026). CRAVE instead induces recurrent visual–state prototypes, orders them by corpus-level occurrence and decodes each complete episode before ranking future progress increments. The distinction is therefore framewise correction of a time proxy versus whole-episode induction of a latent progress coordinate. Because UR-VC also changes the encoder, retrieval rule and binary interface, we treat it as a source-native diagnostic rather than a label-source-matched comparator.

Multimodal task structure. R3M and VIP learn reusable visual geometry (Nair et al., 2023; Ma et al., 2023), while manipulation-centric pretraining also encodes robot state and action (Jiang et al., 2025). Frozen DINOv2, DINOv3 and SigLIP 2 features (Oquab et al., 2024; Siméoni et al., 2025; Tschannen et al., 2025) still do not assign task order. Temporal alignment and procedural discovery show that repeated videos contain ordered structure across changes in speed and viewpoint (Sermanet et al., 2018; Dwibedi et al., 2019; Haresh et al., 2021; Dvornik et al., 2023; Shah et al., 2023). CRAVE converts this structure into an episode-coupled progress coordinate and ordinal policy conditions; its prototypes are recurrent observation–state neighbourhoods, not semantic stages, rewards or calibrated state values.

3 Problem Setup and Policy Interface

For one task, let $\mathcal{D} = \{\tau_i\}_{i=1}^{N}$ contain N demonstration episodes. Episode i has length T_i and comprises observations $o_{i,t} \in \mathcal{O}$, robot states $x_{i,t} \in \mathbb{R}^{d_x}$ and normalized action chunks $a_{i,t} \in \mathbb{R}^{d_a}$ for $t \in \{1, \dots, T_i\}$. Demonstrations contain neither a ground-truth reward nor a calibrated policy value. CRAVE maps each complete episode to a normalized progress path $p_i : \{1, \dots, T_i\} \to [0, 1]$. For an integer horizon $H > 0$, it defines the clipped finite-horizon progress increment

$$\Delta p_{i,t} = p_i(\min(t + H, T_i)) - p_i(t). \tag{1}$$

In the folding experiments, observations are sampled at 30 Hz and $H = 50$ frames, corresponding to 1.67 s. A fixed quantizer $Q : [0, 1] \to \{\text{LOW}, \text{MIDDLE}, \text{HIGH}\}$ maps the empirical percentile rank of $\Delta p_{i,t}$ within the training set to an ordinal condition $y_{i,t}$. These conditions describe relative progress increments, not semantic stages, rewards or numerical advantages.

For a label source $s \in \{\text{CRAVE}, \chi_0\text{-AE}\}$, write $y_{i,t}^{(s)}$ for the corresponding ordinal label. Let $\epsilon \sim \mathcal{N}(0, I_{d_a})$, where I_{d_a} is the d_a-dimensional identity matrix, sample flow time $u \sim \mathcal{U}(0, 1)$, and set $a_u = u\epsilon + (1 - u)a_{i,t}$. Every conditioned $\pi_{0.5}$ arm minimizes

$$\mathcal{L}_{\text{FM}}^{(s)} = \mathbb{E}\left[\left\| v_\theta(a_u, u, o_{i,t}, q \oplus \phi(y_{i,t}^{(s)})) - (\epsilon - a_{i,t}) \right\|_2^2\right], \tag{2}$$

where the expectation is over training rows (i, t), Gaussian noise ϵ and flow time u; v_θ is the conditional flow field with parameters θ; q is the task instruction; ϕ maps an ordinal label to a fixed text suffix; and $\oplus$ denotes text concatenation. Plain BC uses q without a suffix. Direct CRAVE changes only the condition attached to each training row; the action targets, flow-matching loss and deployed policy architecture remain unchanged. This is the flow-matching training objective used by the $\pi_0/\pi_{0.5}$ policy family (Lipman et al., 2023; Black et al., 2025; Physical Intelligence et al., 2025b).

4 CRAVE

CRAVE converts repeated task demonstrations into ordinal conditions through three offline operations (Fig. 2). It discovers configurations that recur across episodes, decodes those configurations into one progress path per episode, and relabels each policy row by its future progress increment. Here, recurrent denotes support across independently collected episodes rather than a recurrent neural-network architecture. The complete episode is available during relabeling; at deployment the robot runs the post-trained $\pi_{0.5}$ policy alone.

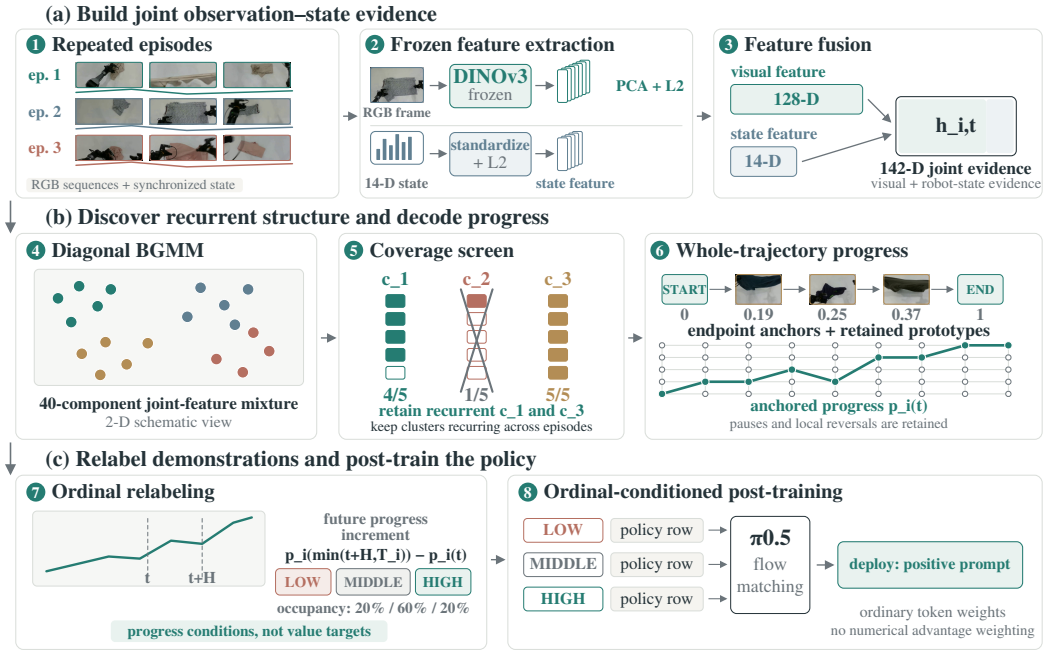

Figure 2: CRAVE transforms repeated demonstrations into policy conditions. **(a)** Frozen visual and robot-state features from repeated episodes are concatenated into joint evidence. **(b)** Mixture components are screened by cross-episode coverage; the retained prototypes and endpoint anchors are decoded jointly into a progress path that permits pauses and local reversals. **(c)** Finite-horizon progress increments are ranked into ordinal prompt conditions that train the unchanged $\pi_{0.5}$ objective with ordinary token weights. In the schematic, c_k denotes a mixture component and $r/5$ denotes support in r of five example episodes.

4.1 Discover recurrent cross-episode structure

Let $f : \mathcal{O} \to \mathbb{R}^{d_v}$ be a frozen visual encoder and $P \in \mathbb{R}^{128\times d_v}$ the principal component analysis (PCA) projection fitted on training frames. Define $\mathrm{L2}(z) = z/(\|z\|_2 + 10^{-9})$. Training-set state mean $\mu_x \in \mathbb{R}^{14}$ and stabilized standard deviation $\tilde{\sigma}_x = \sigma_x + 10^{-6} \in \mathbb{R}^{14}$ map each 14-dimensional robot state to the joint representation

$$h_{i,t} = \left[\mathrm{L2}(P\,\mathrm{L2}(f(o_{i,t}))), \mathrm{L2}((x_{i,t} - \mu_x) \oslash \tilde{\sigma}_x)\right], \tag{3}$$

where $\oslash$ is elementwise division and $h_{i,t} \in \mathbb{R}^{142}$. The garment-folding implementation uses frozen DINOv3 features (Siméoni et al., 2025) and robot state; action and velocity are not inputs to this representation. The state order is six left-arm joints, left gripper, six right-arm joints and right gripper. Visual features associate similar configurations across episodes, while robot state separates contact states that appear alike from the cameras.

CRAVE fits a K-component diagonal Bayesian Gaussian mixture model (BGMM) to the joint features and counts support by episode, not by frame. Let $\mathcal{D}_{\mathrm{strata}}$ index collection strata. For stratum $d \in \mathcal{D}_{\mathrm{strata}}$, let N_d be its number of episodes and $n_{k,d}$ the number containing at least one frame

assigned to component $k \in \{1, \ldots, K\}$. Component k is retained when, for at least one stratum, its episode coverage exceeds $\rho \in (0, 1]$ and its support exceeds the integer threshold m_d:

$$\exists d \in \mathcal{D}_{\text{strata}} : \quad n_{k,d}/N_d \geq \rho \quad \wedge \quad n_{k,d} \geq m_d. \tag{4}$$

This screen prevents long dwell times or duplicated frames in one episode from creating a prototype. For each retained component, the ℓ_2-normalized component mean becomes a recurrent prototype c_k, and the median normalized position t/T_i of its assigned frames provides its order coordinate $b_k \in [0, 1]$. A prototype represents a recurrent observation–state neighbourhood rather than a human-named stage.

4.2 Decode episode-level progress

Independent nearest-prototype assignment can switch rapidly between visually similar contact states. CRAVE instead explains all frames in an episode jointly. Start and end anchors are the normalized means of the first and last three joint features in the training episodes and receive coordinates 0 and 1. They are inserted as prototypes in $\mathcal{C}_0$ and $\mathcal{C}_1$, respectively. Let $B = \text{sort}(\{0, 1\} \cup \{b_k\})$ be the ordered coordinate set, and let $\mathcal{C}_b$ contain all prototypes assigned to coordinate $b \in B$. For transition penalty $\lambda \geq 0$, the emission and transition costs are

$$e_{i,t}(b) = \min_{c \in \mathcal{C}_b} \|h_{i,t} - c\|_2, \qquad A(b', b) = \lambda |b - b'|, \tag{5}$$

and the decoded path is

$$b^*_{i,1:T_i} = \arg \min_{b_{1:T_i} \in B^{T_i}} \left[\sum_{t=1}^{T_i} e_{i,t}(b_t) + \sum_{t=2}^{T_i} A(b_{t-1}, b_t) \right], \quad b_1 = 0,\ b_{T_i} = 1. \tag{6}$$

Here $e_{i,t}(b)$ is the distance from frame feature $h_{i,t}$ to the nearest prototype at coordinate b, and $A(b', b)$ penalizes a coordinate change from b' to b. All-to-all transitions allow phase skipping and backward motion; the anchors fix only the episode endpoints. Equal-cost paths are resolved by the deterministic ascending order of coordinates in B. Within each contiguous decoded segment, CRAVE retains the frame nearest to the selected prototype. It linearly interpolates the retained frame coordinates to obtain $p_i(t)$. This whole-episode decoder can represent pauses, re-grasps and local regressions without forcing a monotone path.

4.3 Relabel demonstrations for policy post-training

Equation 1 converts the decoded path into one progress increment per policy row. CRAVE ranks increments within the current training set and maps the bottom 20%, middle 60% and top 20% to LOW, MIDDLE and HIGH. This occupancy-matched interface is designed for the shape of the recovered CRAVE progress increments; it is not presented as a natural three-stage ontology. The same occupancy is applied to outputs from the Stage Advantage estimator of χ_0 (Yu et al., 2026) in the matched policy study. We refer to that adapted label source as χ_0-AE, where AE denotes the advantage estimator in our tables. Thus label source changes without changing prompt frequency. The conditions mean low, medium and high progress increment relative to a route and dataset; they do not assert regression, stasis and advancement as absolute semantic classes.

The conditions are materialized as the historical text suffixes `negative`, `normal` and `positive`. Policy training uses Eq. 2 with ordinary token weights and no numerical advantage weighting. Conditioned policies deploy with the fixed upper-score prompt. Appendix A.2 specifies the projection, mixture, recurrence, decoding, quantization and export settings used in the experiments.

5 Experiments

5.1 Experimental setup

Policy routes and data. The primary study used nested garment-folding sets $D_{150} \subset D_{300} \subset D_{450}$ drawn from one 464-episode source. Each set trained three routes from the same $\pi_{0.5}$ initialization: plain BC, χ_0-AE conditioning from a fixed historical Stage Advantage estimator checkpoint (Yu et al., 2026), and Direct CRAVE. Within each scale, policy rows, actions, augmentation,

optimizer, 40,000-update budget, checkpoint rule and deployment prompt were fixed. CRAVE refit its offline structure to each set; χ_0-AE applied the same pretrained estimator to the corresponding policy rows and then used our matched 20/60/20 ordinal interface. This comparator isolates label source; it is not a reproduction of every module in χ_0. A source-native policy checkpoint trained with UR-VC-derived labels at D_{450} serves as a separate diagnostic rather than a tenth matched arm.

Nail painting provides a second physical comparison between plain $\pi_{0.5}$ SFT and Direct CRAVE. Both policies used 283 training episodes, 71 validation episodes, the same initialization, seed, batch size and 40,000-update schedule. An outcome-independent rule selected 20 evaluation episodes per route. Completion is scored over four ordered subtasks, each worth 25 percentage points.

The writing study uses a separate 61-episode single-arm corpus collected by four operators and deterministically split into 55 training and six validation episodes. Direct CRAVE starts from the same $\pi_{0.5}$ base checkpoint used throughout the study and trains for 40,000 updates with seed 42 and a global batch of 128. The policy observes only the top-head camera and controls the seven-dimensional left arm. A rollout succeeds only when the ordered I/C/L/R sequence is completed.

Robot protocol and endpoints. Folding policies ran on a Piper dual-arm robot with two wrist D405 cameras, a head D435 camera and zero-joint initialization. Every arm used 20 sequential rollouts, balanced over four garment colours. Binary task success is the primary endpoint. We also report the mean observed duration over all recorded rollouts and median [interquartile range] among successful rollouts. These durations describe robot occupancy directly and are not adjusted for outcome. The independent unit is one rollout. Wilson score intervals (Wilson, 1927) condition on the single deployed checkpoint for each arm. The rollout blocks were sequential rather than randomized or blinded, so policy comparisons are descriptive and do not capture training-seed variation. Full protocol, reconciliation and sensitivity records appear in Appendix A.5.

Offline and optimization metrics. The structural diagnostic compares each decoded progress trace with one human flatten-completion boundary per episode. A progress threshold is selected on a separate development split from the grid $\{0.01, \ldots, 0.99\}$ to minimize mean absolute boundary error. On a held-out episode, the predicted boundary is the first frame at which progress crosses that frozen threshold, or the nearest-threshold frame if no crossing occurs. Boundary error is the absolute frame displacement divided by 30 Hz and is averaged over episodes. The reported reduction is comparator error minus full-CRAVE error; intervals use 10,000 paired episode-bootstrap samples.

For policy optimization, $\mathrm{MAE@}h$ is the mean absolute difference between the first h predicted and ground-truth future actions, averaged over the 14 controlled dimensions and then over frozen validation query rows. $\mathrm{MAE@1}$ therefore evaluates only the first predicted action. We evaluate every 2,000 updates through step 38,000 and once at step 39,999. The threshold endpoint is the first observed grid point with $\mathrm{MAE@1} < 0.01$. Normalized curve area is the trapezoidal integral of $\mathrm{MAE@1}$ over steps 2,000–39,999 divided by the covered step span; it is consequently the step-weighted mean validation error over the observed training trajectory.

5.2 CRAVE MATCHES OR IMPROVES FOLDING PERFORMANCE ACROSS DATA SCALES

Direct CRAVE produced the highest or joint-highest observed success count at all three training-set sizes (Table 1). It completed 11/20, 15/20 and 16/20 rollouts at D_{150}, D_{300} and D_{450}, respectively. Plain BC completed 10/20, 11/20 and 16/20; χ_0-AE completed 8/20, 14/20 and 13/20. The corresponding CRAVE-minus-BC differences were +5, +20 and 0 percentage points, while CRAVE-minus-χ_0-AE differences were +15, +5 and +15 points. Their rollout-level confidence intervals are reported in Appendix A.7.

The timing measurements distinguish policies that reached the same terminal outcome with different robot occupancy. CRAVE's all-rollout means were 86.0, 74.7 and 75.4 s, below χ_0-AE at every scale. Its successful-rollout median fell to 58.1 s at D_{300} and 55.8 s at D_{450}, the shortest matched medians in those cells. One successful D_{150} CRAVE rollout lacks recoverable timing metadata. Figure 3 shows success and absolute mean duration with a single visual emphasis on CRAVE.

Table 1: Garment-folding policy utility for the nine primary single-seed checkpoints. Success is successes/20 rollouts [Wilson 95% CI]. Time columns report the unadjusted all-rollout mean (n recorded) and successful-rollout median [IQR]; one successful D150-CRAVE duration is missing. Bold marks the observed best within a demonstration scale; ties do not imply significance.

Demonstrations	Policy route	Success (95% CI) ↑	Mean duration (s; n) ↓	Successful median [IQR] (s) ↓
150	Plain $\pi_{0.5}$ BC	10/20; 50.0% [29.9–70.1]	96.2; 20	71.6 [59.1–104.8]
	χ_0-AE conditioned	8/20; 40.0% [21.9–61.3]	111.9; 20	**61.9 [56.3–95.6]**
	Direct CRAVE	**11/20; 55.0% [34.2–74.2]**	**86.0; 19**	76.9 [63.7–102.2]
300	Plain $\pi_{0.5}$ BC	11/20; 55.0% [34.2–74.2]	78.0; 20	60.4 [43.8–77.8]
	χ_0-AE conditioned	14/20; 70.0% [48.1–85.5]	109.1; 20	91.4 [72.8–120.1]
	Direct CRAVE	**15/20; 75.0% [53.1–88.8]**	**74.7; 20**	**58.1 [53.1–91.1]**
450	Plain $\pi_{0.5}$ BC	**16/20; 80.0% [58.4–91.9]**	**71.2; 20**	57.8 [44.5–83.6]
	χ_0-AE conditioned	13/20; 65.0% [43.3–81.9]	89.3; 20	60.1 [55.3–74.9]
	Direct CRAVE	**16/20; 80.0% [58.4–91.9]**	75.4; 20	**55.8 [44.3–80.4]**

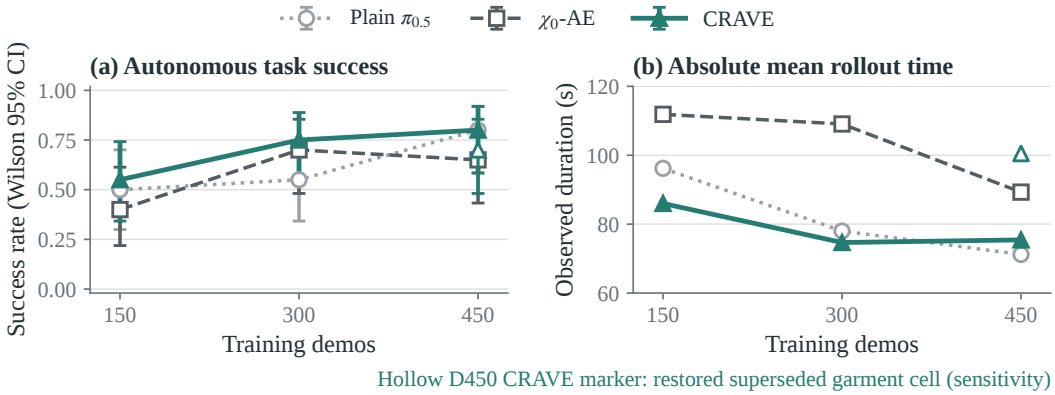

Figure 3: Garment-folding results across nested demonstration sets. **(a)** Success in 20 sequential rollouts per checkpoint with Wilson 95% intervals; the hollow D_{450} CRAVE marker restores the superseded garment block as a sensitivity analysis. **(b)** Unadjusted all-rollout mean duration. CRAVE is teal and matched comparators are grey.

At D_{450}, Direct CRAVE, plain BC and the separate source-native UR-VC diagnostic each completed 16/20 rollouts; UR-VC's mean observed duration was 84.8 s. CRAVE therefore reached the same observed success count without UR-VC's retrieval and binary prompt interface. Appendix A.5 reports all ten checkpoints.

5.3 Whole-trajectory decoding recovers coherent progress

On 300 held-out folding episodes, whole-trajectory decoding reduced boundary error by 12.72 s relative to framewise assignment (95% paired-bootstrap interval, 11.38–14.09 s). Removing the global decoded path increased the difference to 17.59 s (16.26–18.94 s). These matched comparisons isolate the benefit of coupling assignments across an episode (Table 2).

Table 2: Matched structural ablations on 300 duration-balanced held-out folding episodes. Values are comparator-minus-CRAVE boundary-error differences with paired-bootstrap 95% intervals.

Removed structure	Independent units	Boundary-error reduction (s) ↑
Replace whole-trajectory decoding with framewise assignment	300 episodes	12.72 [11.38, 14.09]
Remove the decoded global path	300 episodes	17.59 [16.26, 18.94]

The decoded path also responds to local recovery. Around human takeover in six DAgger episodes (Ross et al., 2011), Direct CRAVE's event-specific five-second slope lift was positive in five episodes (mean 0.0081 normalized progress per second; 95% episode-bootstrap interval, -0.0159 to 0.0344; Fig. 4b). This trajectory view is descriptive; the boundary experiment provides the direct mechanism test. A matched learned-probe diagnostic is reported in Appendix A.14; the source-native UR-VC policy diagnostic is reported separately above because it does not share the matched label interface.

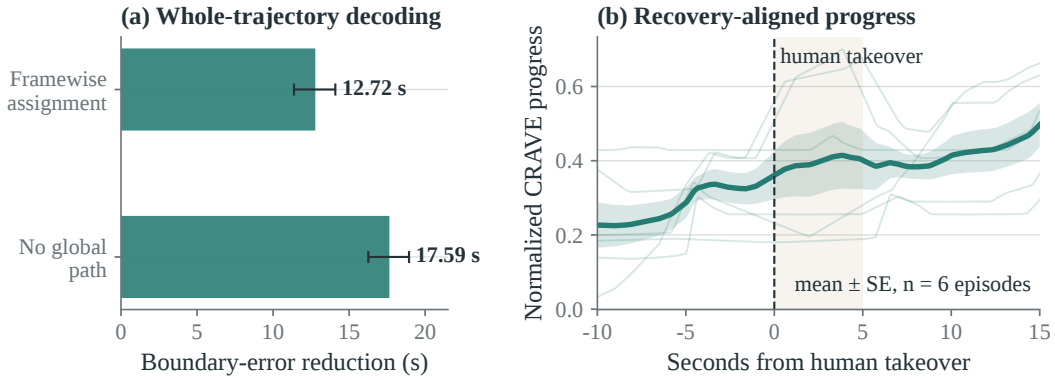

Figure 4: Structural evidence for recovered progress. **(a)** Mean boundary-error reduction of full CRAVE relative to framewise assignment and removal of the decoded global path on 300 held-out folding episodes; error bars are paired-bootstrap 95% intervals. **(b)** Direct CRAVE progress around human takeover in six held-out DAgger sequences (Ross et al., 2011); thin curves are episodes and the thick curve is the mean with standard error. Progress is normalized within episode for visualization, and the shaded interval is the first five seconds after takeover.

5.4 Progress relabeling transfers to nail painting and low-data writing

On the preselected 20-episode subsets, Direct CRAVE raised mean four-stage completion from 30% to 35% (Table 3). Every CRAVE rollout completed at least one stage, compared with 18/20 SFT rollouts, and 8/20 CRAVE rollouts completed two stages, compared with 6/20 for SFT. These are task-completion scores, not binary success rates; the study therefore supports transfer of the relabeling route without pooling nail-painting and folding outcomes.

Table 3: Nail-painting completion on the preselected 20-episode subset. Each episode receives 25 percentage points for each of four ordered subtasks completed. Columns 0/25/50% give the exact score distribution; threshold columns report episodes completing at least one or two stages. Neither route reached 75% or 100% completion. This is a completion score, not a binary success rate. Bold denotes the observed larger count or mean.

Policy route	n	0%	25%	50%	$\geq$1 stage	$\geq$2 stages	Mean ↑
Plain $\pi_{0.5}$ SFT	20	2	12	6	18/20	6/20	30%
Direct CRAVE	20	0	12	8	**20/20**	**8/20**	**35%**

For ordered writing, Direct CRAVE completed 16/19 unique robot rollouts (84.2%; Wilson 95% confidence interval, 62.4–94.5%) after training on 55 demonstrations from four operators; one repaired byte-identical duplicate was excluded. This extends the same relabeling and training interface to low-data single-arm precision writing.

5.5 Label construction removes the learned-estimator stage

For 450 folding demonstrations, CRAVE automatic label construction required a projected 3.7–5.6 min, versus 66–137 min for inference and export from an existing χ_0-AE checkpoint (Fig. 5). CRAVE has no human progress-annotation or learned-estimator-training stage, represented by zeros in the figure. Because the entries mix engineering projections and a workload proxy, stages are

reported separately rather than summed into an end-to-end speedup. The matched claim is limited to automatic label construction.

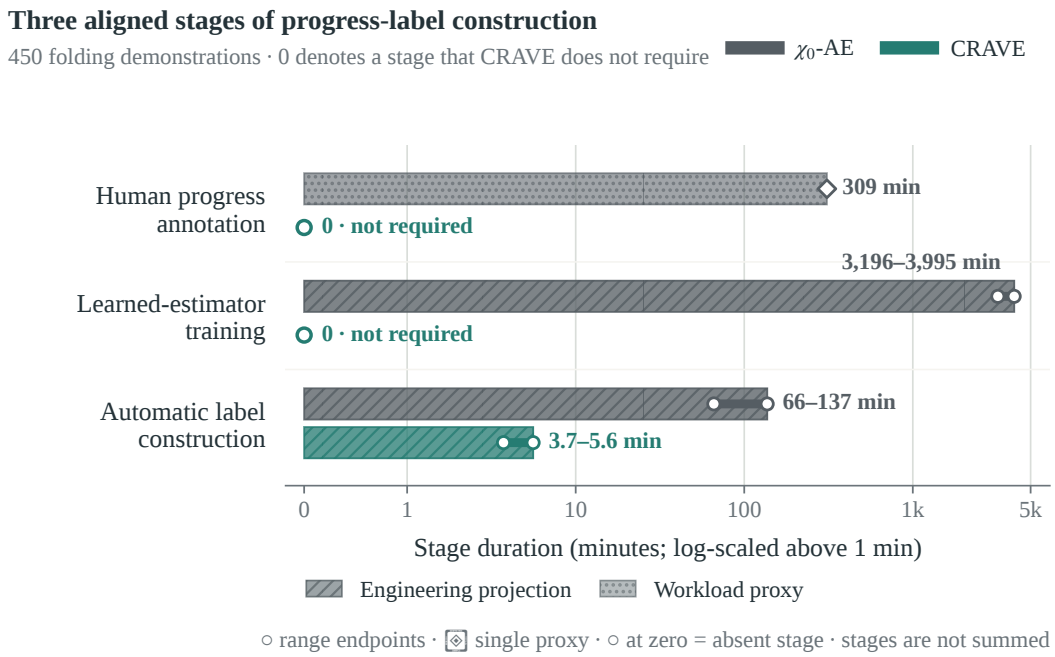

Figure 5: Stage-decomposed cost for 450 folding demonstrations. Zero denotes an absent stage; texture separates evidence classes; open-circle segments show projection ranges and diamonds show single proxies. Ranges are not confidence intervals, stages are not summed, and shared data collection, policy post-training and deployment are excluded.

6 Conclusion and Limitations

CRAVE turns repeated cross-episode structure into progress conditions for VLA post-training. Whole-trajectory decoding improved boundary localization, and Direct CRAVE achieved the highest or joint-highest folding success count at all three data scales while eliminating the learned-estimator stage. Nail painting, 16/19 completed writing rollouts and minute-scale label construction extend the evidence across tasks and operational cost. These robot results use one checkpoint per arm and sequential, non-blinded rollouts, so their intervals capture rollout rather than training-seed variation. Within this offline complete-episode setting, CRAVE removes phase annotation and a separate reward, value or advantage estimator without changing the deployed policy; online and cross-embodiment recovery remain future tests.

AI use statement

Generative AI tools supported literature review, methodological and implementation feedback, analysis review, and manuscript preparation, including translation. They did not generate the robot data, rollout labels, or reported measurements. The authors reran the reported analyses and builds, verified claims against frozen source artifacts, and take responsibility for the final submission.

Reproducibility statement

Section 5 specifies the data, policy, checkpoint, robot, and statistical contracts used in the main comparison. Appendix A records the exact label interface, phase-induction and decoding parameters, per-arm results, provenance limitations, and source-artifact pointers. The anonymous supplementary package includes the executed training and relabeling contracts, sanitized paper-facing robot and offline records, immutable source hashes, and a deterministic audit that verifies package in-

tegrity and recomputes the reported headline invariants. Raw videos, full robot corpora, and model weights are excluded from the anonymous artifact.

REFERENCES

Kevin Black, Noah Brown, Danny Driess, et al. π_0: A vision-language-action flow model for general robot control. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.010. URL https://www.roboticsproceedings.org/rss21/p010.html.

Anthony Brohan, Noah Brown, Justice Carbajal, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics*, 3(1):1–27, 1974. doi: 10.1080/03610927408827101.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. In *Advances in Neural Information Processing Systems*, volume 34, pp. 15084–15097, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/7f489f642a0ddb10272b5c31057f0663-Abstract.html.

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 XIX*, 2023. doi: 10.15607/RSS.2023.XIX.026. URL https://roboticsproceedings.org/rss19/p026.html.

David L. Davies and Donald W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227, 1979. doi: 10.1109/TPAMI.1979.4766909.

Nikita Dvornik, Isma Hadji, Ran Zhang, Konstantinos G. Derpanis, Richard P. Wildes, and Allan D. Jepson. Stepformer: Self-supervised step discovery and localization in instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18952–18961, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Dvornik_StepFormer_Self-Supervised_Step_Discovery_and_Localization_in_Instructional_Videos_CVPR_2023_paper.html.

Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1801–1810, 2019. URL https://openaccess.thecvf.com/content_CVPR_2019/html/Dwibedi_Temporal_Cycle-Consistency_Learning_CVPR_2019_paper.html.

Scott Emmons, Benjamin Eysenbach, Ilya Kostrikov, and Sergey Levine. RvS: What is essential for offline RL via supervised learning? In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=S874XAIpkR-.

Sanjay Haresh, Sateesh Kumar, Huseyin Coskun, Shahram N. Syed, Andrey Konin, Zeeshan Zia, and Quoc-Huy Tran. Learning by aligning videos in time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5548–5558, 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Haresh_Learning_by_Aligning_Videos_in_Time_CVPR_2021_paper.html.

Guangqi Jiang, Yifei Sun, Tao Huang, Huanyu Li, Yongyuan Liang, and Huazhe Xu. Robots pre-train robots: Manipulation-centric robotic representation from large-scale robot datasets. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=yTEwmr1TJb.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, et al. OpenVLA: An open-source vision-language-action model. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713, 2025. URL https://proceedings.mlr.press/v270/kim25c.html.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit Q-learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=68n2s9ZJWF8.

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

Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. VIP: Towards universal visual reward and representation via value-implicit pre-training. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=YJ7o2wetJ2.

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 1678–1690, 2022. URL https://proceedings.mlr.press/v164/mandlekar22a.html.

Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. AWAC: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020. URL https://arxiv.org/abs/2006.09359.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3M: A universal visual representation for robot manipulation. In *Proceedings of the 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 892–909, 2023. URL https://proceedings.mlr.press/v205/nair23a.html.

Robert G. Newcombe. Interval estimation for the difference between independent proportions: Comparison of eleven methods. *Statistics in Medicine*, 17(8):873–890, 1998. doi: 10.1002/(SICI)1097-0258(19980430)17:8<873::AID-SIM779>3.0.CO;2-I.

Franki Nguimatsia Tiofack, Théotime Le Hellard, Fabian Schramm, Nicolas Perrin-Gilbert, and Justin Carpentier. Guided flow policy: Learning from high-value actions in offline reinforcement learning. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=EBjy1rmpv0.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, et al. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.090. URL https://www.roboticsproceedings.org/rss20/p090.html.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=a68SUt6zFt.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. URL https://www.jmlr.org/papers/v12/pedregosa11a.html.

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. URL https://arxiv.org/abs/1910.00177.

Physical Intelligence, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Jared DiCarlo, et al. $\pi^*_{0.6}$: A VLA that learns from experience. *arXiv preprint arXiv:2511.14759*, 2025a. URL https://arxiv.org/abs/2511.14759.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025b. URL https://arxiv.org/abs/2504.16054.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635, 2011. URL https://proceedings.mlr.press/v15/ross11a.html.

Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987. doi: 10.1016/0377-0427(87)90125-7.

Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, and Sergey Levine. Time-contrastive networks: Self-supervised learning from video. In *IEEE International Conference on Robotics and Automation*, pp. 1134–1141, 2018. doi: 10.1109/ICRA.2018.8462891. URL https://arxiv.org/abs/1704.06888.

Anshul Shah, Benjamin Lundell, Harpreet Sawhney, and Rama Chellappa. STEPs: Self-supervised key step extraction and localization from unlabeled procedural videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10375–10387, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Shah_STEPs_Self-Supervised_Key_Step_Extraction_and_Localization_from_Unlabeled_Procedural_ICCV_2023_paper.html.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025. URL https://arxiv.org/abs/2508.10104.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025. URL https://arxiv.org/abs/2502.14786.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008. URL https://jmlr.org/papers/v9/vandermaaten08a.html.

Edwin B. Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927. doi: 10.1080/01621459.1927.10502953.

Wenli Xiao, Haotian Lin, Andy Peng, Haoru Xue, Tairan He, Yuqi Xie, Fengyuan Hu, Jimmy Wu, Zhengyi Luo, Linxi Fan, Guanya Shi, and Yuke Zhu. Self-improving vision-language-action models with data generation via residual RL. *arXiv preprint arXiv:2511.00091*, 2025. URL https://arxiv.org/abs/2511.00091.

Checheng Yu, Chonghao Sima, Gangcheng Jiang, Hai Zhang, Haoguang Mai, Hongyang Li, Huijie Wang, Jin Chen, Kaiyang Wu, Li Chen, Lirui Zhao, Modi Shi, Ping Luo, Qingwen Bu, Shijia Peng, Tianyu Li, and Yibo Yuan. χ_0: Resource-aware robust manipulation via taming distributional inconsistencies, 2026. URL https://arxiv.org/abs/2602.09021.

Lirui Zhao, Modi Shi, Li Chen, Qi Liu, Ping Luo, and Hongyang Li. UR-VC: Unsupervised robotic value correction for time-derived progress proxies, 2026. URL https://arxiv.org/abs/2607.12892.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems XIX*, 2023. doi: 10.15607/RSS.2023.XIX.016. URL https://roboticsproceedings.org/rss19/p016.html.

A Appendix

A.1 Executed relabeling algorithm

1. **Represent frames.** Fit PCA-128 and state normalization on D_k; concatenate normalized frozen DINOv3 and 14-D robot-state features.
2. **Induce recurrent phases.** Fit a 40-component diagonal BGMM. Retain components satisfying Eq. 4; use their normalized means as prototypes and median normalized times as order coordinates.
3. **Decode each episode.** Add three-frame start/end anchors, run all-to-all dynamic programming with fixed endpoints and $A(b', b) = \lambda|b - b'|$, and interpolate between prototype-representative frames to obtain $p_i(t)$.
4. **Materialize conditions.** Compute Eq. 1, rank scores within (D_k, s), and map the lower/middle/upper 20/60/20% to the legacy prompt tokens.

The executed artifacts hash the source data, representation, prototypes, decode, score, and quantizer outputs separately.

Table 4: Main-method notation used by the progress-recovery and policy objectives.

Symbol	Definition
i, t, T_i, N	episode index, frame index, episode length and number of episodes
$\mathcal{O}, d_x, d_a, d_v$	observation space and state, action and visual-feature dimensions
$o_{i,t}, x_{i,t}, a_{i,t}$	observation, robot state and normalized action chunk
$f, P, h_{i,t}$	frozen visual encoder, train-fit PCA projection and 142-D joint feature
$\mu_x, \sigma_x, \tilde{\sigma}_x, \oslash$	state mean, standard deviation, stabilized deviation and elementwise division
K, k, c_k, b_k	mixture size, component index, retained prototype and order coordinate
$\mathcal{D}_{\text{strata}}, d, N_d, n_{k,d}, \rho, m_d$	collection strata, stratum index and size, episode support, coverage threshold and minimum support
$B, \mathcal{C}_b, e_{i,t}, A, \lambda$	order set, prototypes at an order, emission cost, transition cost and penalty
$b^*_{i,t}$	decoded order coordinate for frame t of episode i
$p_i(t), H, \Delta p_{i,t}$	recovered progress, future horizon and progress increment
$Q, y^{(s)}_{i,t}, \phi, q, \oplus$	rank quantizer, ordinal label, prompt-suffix map, task instruction and text concatenation
$u, \epsilon, I_{d_a}, a_u, v_\theta$	flow time, Gaussian noise, identity matrix, interpolated action and conditional flow field

A.2 Implementation details

Table 5: Garment-folding relabeling configuration. These settings define the reported route; parameter selection is not a contribution. PCA and BGMM use scikit-learn (Pedregosa et al., 2011).

Component	Setting
Visual representation	frozen DINOv3-base pooled feature
Visual projection	train-fit PCA, 128-D; 20,000-frame fit; seed 0; L2 normalized
Robot evidence	left joints 1–6, left gripper, right joints 1–6, right gripper
State normalization	train-set mean/std; 10^{-6} std stabilizer; L2 10^{-9} stabilizer
Action / velocity features	excluded from the folding teacher
Phase model	40-component diagonal BGMM; 80,000-frame fit; seed 0
BGMM fit	120 iterations; Dirichlet-process prior 10^{-2}; scikit-learn 1.7.2
Recurrence gate	date-stratified episode coverage; $\rho = 0.5$
Minimum support	$\min(20, \max(5, \lfloor N_d/4 \rfloor))$ episodes
Episode anchors	mean of first / last three joint features
Decoder	all-to-all dynamic program; $\lambda = 160$
Sampling / progress horizon	30 Hz / $H = 50$ frames = 1.67 s
Ordinal occupancy	lower / middle / upper = 20 / 60 / 20%
Tie breaking	score, episode ID, frame ID
Export tokens	`negative`, `normal`, `positive`
Deployment condition	fixed `Advantage: positive` suffix

The recurrence support threshold in Table 5 is evaluated independently within each collection date. All transforms are fit on the corresponding training subset. Dynamic programming uses $O(T|B|^2)$ time and $O(T|B|)$ memory for T decoded frames and $|B|$ unique order coordinates.

A.3 EXACT ORDINAL INTERFACE

Table 6: Scientific conditions and executed prompt tokens. Tokens encode route-relative score quantiles; conditioned policies deploy with the `positive` token.

Scientific condition	ID	Prompt token
lower score	0	`negative`
middle score	1	`normal`
upper score	2	`positive`

Scores are sorted by $(\text{score}, \text{episode ID}, \text{frame ID})$. The rule is frame-weighted and can split tied values by identity. Every row is retained, including the last H frames whose future index is clipped to the terminal frame. Repeated terminal action tokens are retained in the loss identically across matched arms.

A.4 TRAINING AND DEPLOYMENT CONFIGURATION

Table 7: Common policy-training and robot-deployment contract.

Item	Frozen setting
Initialization	$\pi_{0.5}$
Policy update	full-parameter flow-matching supervised fine-tuning
Update budget	40,000
Global batch / training parallelism	128 / 8 GPUs
Learning-rate schedule	cosine; 500-step warm-up; $10^{-5} \rightarrow 10^{-6}$
Optimizer	AdamW; $\beta = (0.9, 0.95)$; gradient clip 1.0
Weight decay / EMA	10^{-10} / 0.9999
Training seed	42
Validation cadence	every 2,000 steps
Robot	Piper dual-arm
Cameras	left/right D405; head D435
Arm initialization	both arms at zero-joint configuration
CRAVE horizon / transition	$H = 50$ frames (1.67 s at 30 Hz) / $\lambda = 160$
Deployment condition	fixed upper-score (`positive`) prompt

A.5 FULL ROBOT RESULTS AND RECONCILIATION

Table 8: Extended garment-folding policy and optimization summary. Success intervals are Wilson 95% intervals conditional on one checkpoint. All-rollout time is the mean over the indicated duration count; successful time is median [IQR]. MAE@1 and normalized curve area use the positive-prompt 2k-spaced validation trajectory. UR-VC is a separate source-native D450 diagnostic, not a tenth matched arm. One successful D150-CRAVE rollout has no duration, giving 19 all-rollout and 10 successful-duration observations for that row.

Scale	Route	Success (95% CI) ↑	All time (s; n) ↓	Success median [IQR] (s) ↓	First eval $<$ .01 (k steps) ↓	MAE@1_{40k} ↓	Norm. curve area ↓
D150	Plain $\pi_{0.5}$	10/20; 50.0% [29.9–70.1]	96.2; 20	71.6 [59.1–104.8]	30k	0.0092	0.02115
D150	χ_0-AE	8/20; 40.0% [21.9–61.3]	111.9; 20	61.9 [56.3–95.6]	30k	0.0093	0.02130
D150	Direct CRAVE	11/20; 55.0% [34.2–74.2]	86.0; 19	76.9 [63.7–102.2]	30k	0.0092	0.02137
D300	Plain $\pi_{0.5}$	11/20; 55.0% [34.2–74.2]	78.0; 20	60.4 [43.8–77.8]	30k	0.0091	0.02094
D300	χ_0-AE	14/20; 70.0% [48.1–85.5]	109.1; 20	91.4 [72.8–120.1]	30k	0.0091	0.02107
D300	Direct CRAVE	15/20; 75.0% [53.1–88.8]	74.7; 20	58.1 [53.1–91.1]	30k	0.0093	0.02101
D450	Plain $\pi_{0.5}$	16/20; 80.0% [58.4–91.9]	71.2; 20	57.8 [44.5–83.6]	28k	0.0091	0.02067
D450	χ_0-AE	13/20; 65.0% [43.3–81.9]	89.3; 20	60.1 [55.3–74.9]	28k	0.0091	0.02084
D450	Direct CRAVE	16/20; 80.0% [58.4–91.9]	75.4; 20	55.8 [44.3–80.4]	28k	0.0091	0.02087
D450	Source-native UR-VC	16/20; 80.0% [58.4–91.9]	84.8; 20	72.8 [49.1–98.3]	28k	0.0090	0.02082

The enriched table is generated deterministically from the canonical trial ledger, robot aggregate, and convergence summary. Its machine-readable supplementary record stores the three source hashes, duration definitions, independent units, and claim boundary. Training wall times are retained in that artifact for provenance but are not compared because hardware differs across arms and one hardware record is missing.

The canonical ledger retains 210 executed rows. Two hundred enter the balanced analysis; five extra brown F150-BC rows and five superseded original green F450-CRAVE rows remain excluded from the balanced denominator. Of the 210 rows, 209 have raw duration/deployment matches. The unmatched F150-CRAVE T18 outcome is retained, but duration and media are missing without imputation.

Terminal annotations cover 203/210 executed rows and 194/200 primary rows. All covered annotations agree with canonical outcomes after four verified adjudications; seven full-ledger annotations remain missing. The full ledger contains 134 successes and 76 failures. These counts audit outcome reconciliation; an independent second rating was not performed.

For D450-CRAVE, default green trials T21–T25 produced 4/5 successes; original green trials T01–T05 produced 2/5. The canonical balanced summary is 16/20 (80.0%, Wilson 95% CI 58.4–91.9%) with mean observed duration 75.428 s. Restoring the original cell gives 14/20 (70.0%) and 100.513 s. The two cells form separate sensitivity analyses.

A.6 Failure-mode diagnostic

We map the canonical ledger's operator-entered failure text into deterministic primary categories for Table 9: phrases denoting inability to grasp, repeated pinching, or release map to grasp acquisition/retention; remaining strings containing "loop" (case insensitive) map to action loop/out-of-distribution behavior; remaining phrases denoting an untidy fold or incorrect procedure map to fold geometry/sequence; all other strings map to unclassified. This priority assigns the single mixed contact-loss–loop entry to the more specific contact-loss category. The anonymous machine-readable artifact reports the resulting aggregate counts; raw operator-entered strings and private trial paths remain outside the submission package.

Table 9: Deterministic diagnostic collapse of failure reasons in the 200 primary sequential rollouts. D150–D450 aggregate the nine matched Plain $\pi_{0.5}$/χ_0-AE/CRAVE arms, with the source-native UR-VC diagnostic listed separately. Entries are failed-rollout counts.

Set	Failures	Grasp/contact	Loop/out-of-distribution	Geometry/sequence	Unclassified
D150 matched	31	11	14	6	0
D300 matched	20	18	1	0	1
D450 matched	15	13	2	0	0
D450 UR-VC	4	4	0	0	0
All primary	70	46	17	6	1

A.7 Pairwise success contrasts

Table 10: Exploratory absolute risk differences in robot success. Values are CRAVE minus the named comparison route in percentage points, with two-sided 95% Newcombe hybrid-score intervals for two independent proportions (Newcombe, 1998). Intervals quantify rollout variation conditional on the deployed checkpoints; training-seed variation lies outside their scope.

Scale	CRAVE − Plain $\pi_{0.5}$	CRAVE − χ_0-AE
D150	$+5.0$ $[-23.9, +32.8]$	$+15.0$ $[-14.8, +41.4]$
D300	$+20.0$ $[-9.1, +45.0]$	$+5.0$ $[-21.8, +30.9]$
D450	0.0 $[-24.7, +24.7]$	$+15.0$ $[-12.4, +39.8]$

Table 11: Control structure across policy routes. "Same" refers to the corresponding data-scale cell. Source-native UR-VC forms a separate D450 diagnostic alongside the nine-arm matched comparison.

Factor	Plain $\pi_{0.5}$	χ_0-AE	Direct CRAVE	UR-VC diagnostic
Policy rows / actions	Same	Same	Same	D450 source view
Initialization / updates	Same / 40k	Same / 40k	Same / 40k	Same / 40k
Policy objective	Flow matching	Flow matching	Flow matching	Flow matching
Prompt interface	Task only	20/60/20 ordinal	20/60/20 ordinal	Source-native binary
Label source	None	Fixed historical χ_0 Stage Advantage estimator	Current D_k structure	Cross-episode retrieval
Human stage supervision	None	Used to train estimator	None	None
Learned label model	None	Yes, fixed	None	None
Deployment-time labeler	None	None	None	None

A.8 Comparator contract

A.9 Nail-painting annotation and deployment provenance

The nail-painting evaluation labels are linked to two immutable batch manifests. The Direct CRAVE batch ("69cd51bd-7045-496d-8568-0bc85276f2d5") contains 35 labeled episodes from "2026-09-02-v4" and uses the `nail_fullep_viterbi_v3lam24_awbc_bs128_s42_step39999` checkpoint. The SFT batch ("750680af-3c18-4224-be9b-13fdb4a514aa") contains 36 labeled episodes from "2026-08-31-v4" and uses `nail_fullep_sft_pi05base_bs128_s42_step39999`. Every referenced Parquet file and top-head, left-hand and right-hand camera video exists at the source path. Episode 77 in the SFT batch has no deployment field in the source metadata and is retained as checkpoint-provenance missing. The internal audit retains manifest hashes, deployment records, and source paths. The anonymous supplement instead provides source episode IDs, opaque record hashes, the frozen selection rule, and the resulting per-episode scores. For the reported nail-painting metric, each of the four ordered subtasks contributes 25 percentage points to a per-episode completion score. The 30% and 35% values in the main text are aggregate completion scores under this rubric, not episode-success proportions. The canonical 20-episode subset is frozen by sorting each batch by episode ID, dividing it into four temporal quartiles, and selecting local ranks 0, 2, 4, 6 and 7 in every quartile. The resulting sanitized CSV and JSON records are included in the supplementary artifact.

A.10 Low-data writing provenance, comparator and sensitivity

The writing corpus contains 61 cleaned episodes (46,748 frames), with a fixed 55/6 train/validation split and no episode overlap. Four operators contributed the demonstrations; the operator count is author-confirmed, while the legacy source metadata records the shared placeholder `DEVUSER` rather than a per-episode operator identifier. This supports a multi-demonstrator acquisition claim but not held-out-operator generalization. Plain SFT and Direct CRAVE use the same $\pi_{0.5}$ initialization, seed 42, global batch 128, 40,000-update budget and single top-head camera input.

A matched plain-SFT diagnostic used the same train/validation split, initialization, seed, batch size and 40,000-update budget. Plain SFT completed 18/20 rollouts (90.0%; Wilson 95% interval, 69.9–97.2%). Direct CRAVE's audited unique-rollout result was 16/19 (84.2%; 62.4–94.5%). These single-checkpoint results establish neither route superiority nor equivalence.

Table 12: Appendix diagnostic for low-data single-arm writing. Both routes used 55 demonstrations collected by 4 operators, the same train/validation membership, $\pi_{0.5}$ initialization, seed-42 40k-update budget and top-head camera input. Direct CRAVE excludes one repaired byte-identical duplicate; intervals are Wilson 95% intervals over unique robot rollouts.

Policy route	Train demos	Unique rollouts	Successes/rollouts; % [Wilson 95% CI]
Plain $\pi_{0.5}$ SFT	55	20	18/20; 90.0% [69.9–97.2]
Direct CRAVE	55	19	16/19; 84.2% [62.4–94.5]

The annotation batch `45cbb4fc-ddb1-46a4-9757-629f49db1a3c` contains labels for 39 of 40 deployment records. Plain-SFT episode 16 is the missing record and contains no policy command, so the reported 18/20 conservatively retains it as a failure. The nominal Direct-CRAVE result

is 17/20. During dataset repair, CRAVE episode 7 was replaced by a copy of episode 0; their top-head videos are byte-identical. The primary unique-rollout analysis therefore excludes episode 7 and gives 16/19; 84.2% [62.4–94.5]. SFT episode 18 repeats the L-stage annotation and lacks an R-stage annotation, while CRAVE episode 13 lacks an L-stage annotation; episode-level labels are retained and phase coverage remains explicit in the machine-readable aggregate. These results are descriptive for one trained seed and do not establish route superiority or cross-seed robustness.

A.11 Convergence diagnostics

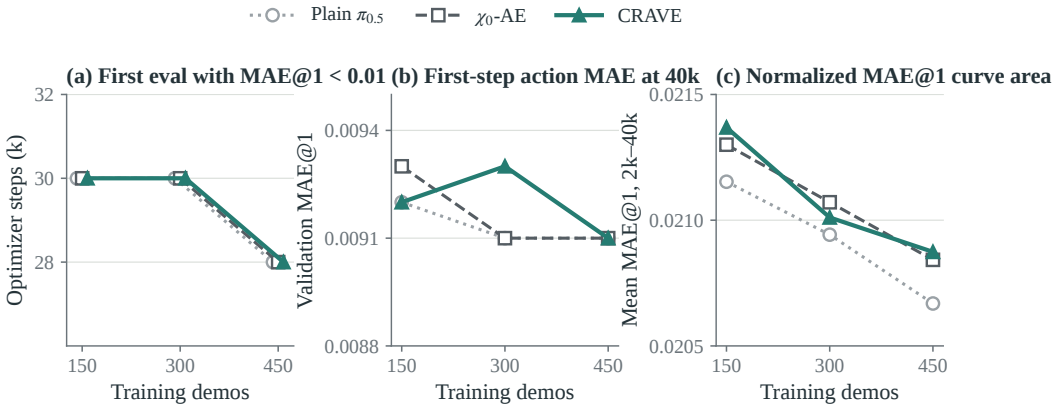

Figure 6: Single-seed positive-prompt optimization diagnostics on the registered 2k-spaced validation grid: first observed step below MAE@1 of 0.01, final MAE@1, and normalized curve area. MAE@1 averages absolute error at the first predicted action over 14 action dimensions and validation query rows; curve area is the trapezoidal integral over steps 2,000–39,999 divided by that step span. The 40k deployment rule is fixed independently. In panel (a), coincident method markers are offset slightly along the horizontal display axis; each triplet still belongs to the labeled demonstration count. The anonymous supplementary artifact includes all 200 validation points and source-log hashes.

All garment-folding arms reached MAE@1 below 0.01 by 30k updates at D150/D300 or 28k at D450. The 0.001 threshold lay beyond the 40k budget for every arm. The adjacent-change criterion $|\Delta\mathrm{MAE@1}| < 2 \times 10^{-4}$ provides an additional curve diagnostic.

A.12 Efficiency evidence classes

Table 13: Stage-cost ledger for the 450-demonstration garment-folding set. Evidence classes are kept separate; no route total is derived. Ranges bracket observed source-log endpoints or the stated equal-GPU-hour scaling sensitivity, not statistical confidence intervals.

Stage	χ_0-AE	CRAVE	Evidence class
Demonstration collection	shared	shared	corpus duration only
Human progress annotation	5.16 h	0 (not required)	one-pass playback proxy
Learned-estimator training	26.6317 h / 16×A100	0 (not required)	historical measured run
Equal-GPU-hour conversion	53.2634–66.5793 h / 8×A100	0 (not applicable)	projection
Automatic label construction	1.10–2.28 h	0.062–0.093 h	8×A100 projection
Policy post-training	28.57 h / 8×H20	25.60 h / 8×A100	cross-hardware logs

The 5.155713 h annotation workload uses the D450 video duration under a one-pass playback assumption. The historical 26.6317 h χ_0-AE estimator-training run used 16 A100 GPUs; the 53.2634–66.5793 h entries project its GPU-hours onto eight A100 GPUs. Cross-hardware policy-loop logs are retained as provenance outside the automatic label-construction component comparison.

A.13 ENCODER AND OFFLINE STRUCTURAL DIAGNOSTICS

The structural diagnostic starts from the frozen held-out pool. For reporting, we form a duration-balanced 300-episode subset by retaining exactly 30 episodes from each of the ten pre-existing duration deciles. In each of the five 31-episode deciles, the retained episodes are the first 30 under a fixed SHA-256 ordering of the episode identifier; predictions, reference labels and outcomes do not enter selection. Phase thresholds remain those selected on the separate development split, and all intervals resample the 300 episodes with 10,000 paired bootstrap replicates. The anonymous supplement records the parent-pool size, selection rule and excluded identifiers.

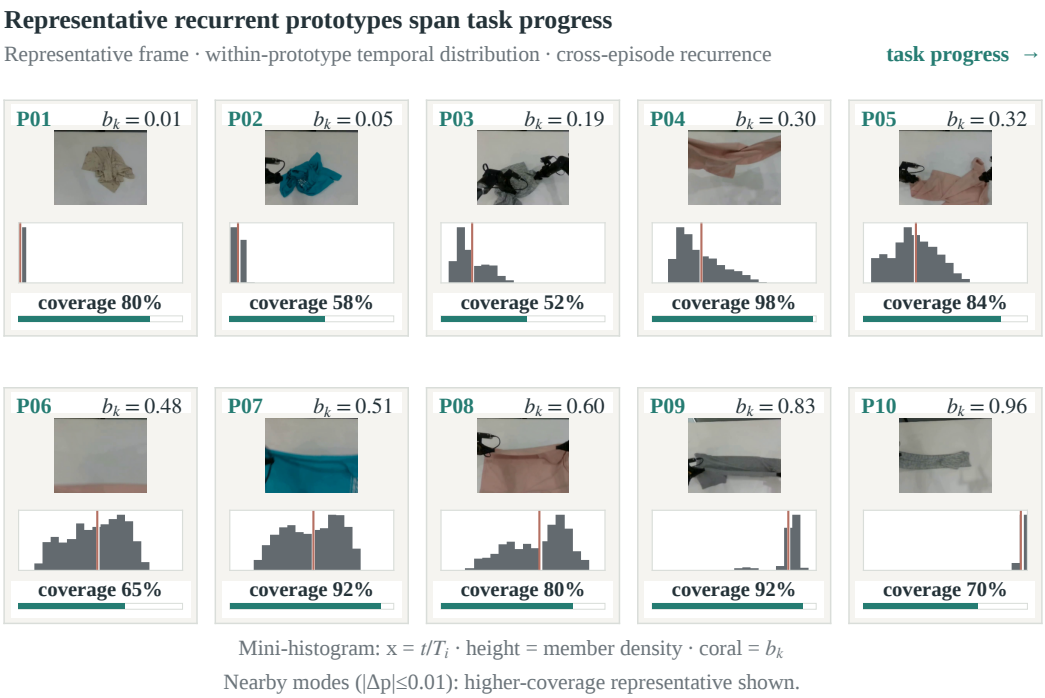

Figure 7: Representative recurrent prototypes from the executed D450 route. Cards are ordered by prototype coordinate b_k, the median normalized episode position, and show a nearest real frame, the member-frame time distribution and cross-episode coverage. Display IDs P01–P10 follow this order. In each mini histogram, the horizontal axis is normalized episode position t/T_i, height is member-frame density, and the coral line marks b_k. When displayed modes lie within 0.01 normalized progress, only the higher-coverage representative is shown; all retained prototypes remain in the decoder.

Silhouette compares within-cluster cohesion with nearest-cluster separation (higher is better); Davies–Bouldin averages each cluster's worst scatter-to- separation ratio (lower is better); and Calinski–Harabasz is a between-cluster to within-cluster dispersion ratio (higher is better) (Rousseeuw, 1987; Davies & Bouldin, 1979; Caliński & Harabasz, 1974).

The matched train-only DINOv2, DINOv3, and SigLIP 2 runs use the same D450 episode membership, sampled frames, preprocessing contract, clustering budget, coverage gate, and event references. They retained 23, 25, and 21 milestones; held-out milestone-nearest silhouettes were 0.0766, 0.0790, and 0.0762, Davies–Bouldin indices were 2.62, 2.54, and 2.77, and Calinski–Harabasz indices were 192.0, 178.4, and 183.1, respectively (Fig. 8). The policy analysis uses DINOv3; matched downstream policy arms would be required to convert these internal-index differences into an encoder-utility ranking.

Table 14: Matched D450 visual-encoder clustering diagnostics. All encoders use the same episodes, sampled frames, robot-state features, BGMM budget, and recurrence gate. Metrics are deterministic single-split internal diagnostics, not downstream policy evaluations. Bold denotes the best observed internal index value; milestone count is descriptive and is not ranked.

Visual encoder	Silhouette ↑	Davies–Bouldin ↓	Calinski–Harabasz ↑	Milestones
DINOv2	0.0766	2.62	**192.0**	23
DINOv3	**0.0790**	**2.54**	178.4	25
SigLIP 2	0.0762	2.77	183.1	21

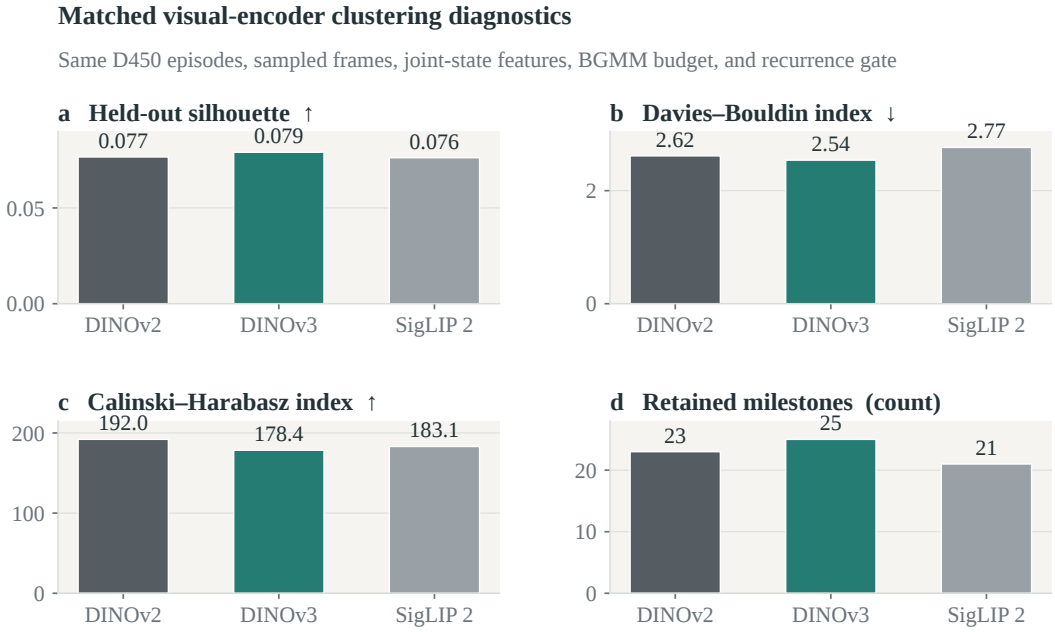

Figure 8: Matched visual-encoder clustering diagnostics on D450. Silhouette, Davies–Bouldin, and Calinski–Harabasz use held-out nearest-milestone assignments on 5,000 frozen held-out frames; retained milestone count is descriptive. All bars are deterministic single-split diagnostics and do not measure downstream policy utility. The indices follow their standard definitions (Rousseeuw, 1987; Davies & Bouldin, 1979; Caliński & Harabasz, 1974). Exact values are reported in Table 14.

A.14 Learned value-estimator failure–recovery diagnostic

Two diagnostic value estimators share architecture, initialization, batch, loss, and a 50k-step budget; only CRAVE progress-path versus human stage-boundary supervision changes. Six held-out DAgger episodes form the independent units. For adjacent five-second windows, let m_{prev}, m_{pre} and m_{human} denote the fitted progress slopes in the intervals $[-10,-5]$, $[-5,0]$ and $[0,5]$ seconds relative to human takeover. Slopes use normalized progress per second. The intervention-specific slope lift is

$$S = (m_{\mathrm{human}} - m_{\mathrm{pre}}) - (m_{\mathrm{pre}} - m_{\mathrm{prev}}). \tag{7}$$

The paired CRAVE-progress-supervised minus human-boundary-supervised lift is 0.2509 (95% episode-bootstrap CI $[0.1666, 0.3386]$). The six paired effects, ordered by frozen held-out episode ID, are 0.150, 0.134, 0.369, 0.179, 0.254, and 0.420; all are positive (exact two-sided paired sign-flip $p = 0.03125$). The direct CRAVE curve is exploratory because its postprocessing saw all D450 episodes, and its interval includes zero. These learned value-estimator probes remain diagnostic and feed neither Direct CRAVE policy training nor deployment.

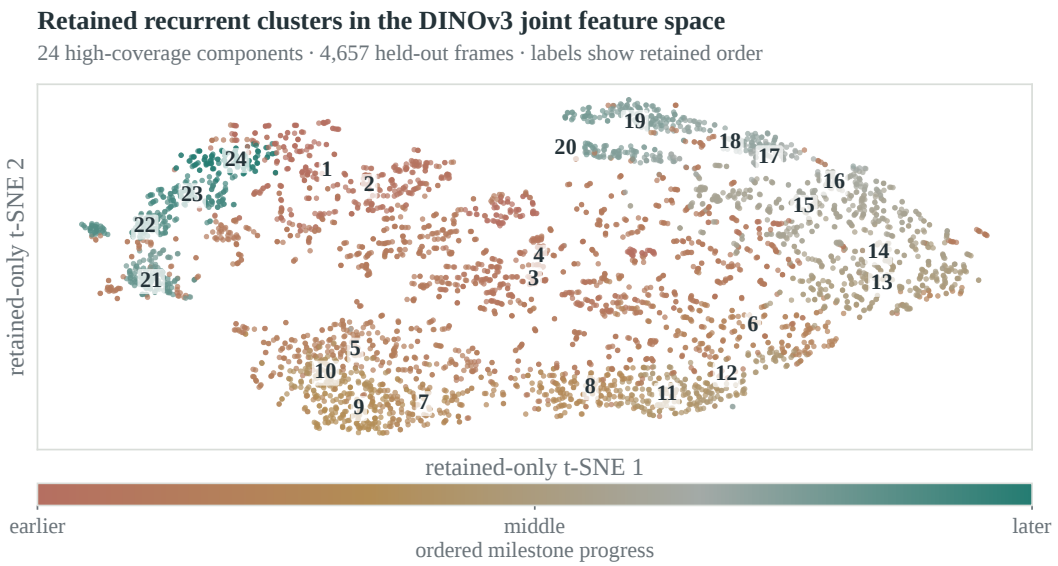

Figure 9: Retained-only t-SNE diagnostic (van der Maaten & Hinton, 2008) for the formal D450 DINOv3 joint feature space. The frozen 5,000-frame held-out sample is filtered before dimensionality reduction: frames outside recurrence-retained components or below 50% overall episode coverage are removed, leaving 4,657 frames from 24 components. The formal recurrence route retains 25 components; this display-only filter removes one. Numeric labels are displayed-component IDs in retained order, not original BGMM component IDs. No unretained candidate points are shown. Colour follows ordered milestone progress. We use perplexity 45, early exaggeration 12, PCA initialization, 1,500 iterations and seed 20260831; t-SNE geometry is qualitative and is not a downstream policy-utility metric.