LEARNING SHARED POSE QUALITY FOR RECURSIVE 6D OBJECT TRACKING

Anonymous authors
Paper under double-blind review

ABSTRACT

Quality-driven decisions in recursive 6D tracking change the hypotheses that the same system must later assess. We study this feedback using a shared model that predicts object-size-normalized ADD-S error for visual observations and temporal priors. Development-calibrated risk determines candidate eligibility, relative predicted error controls fusion, and fresh registration renews an unreliable observation–prior pair. Accepted outputs update motion history, while used registrations reseed the observer. We then refit the quality model once on development candidates encountered under the induced policy. On five post-development sequences from three objects, Full improves sequence-macro AUC by 6.90 points over Matched simple under matched renewal schedules. In a separate four-condition autonomous diagnostic, Full reaches 79.99 versus 58.16 AUC against a development-tuned Autonomous simple policy, but requires 4,477 versus 452 registration calls and does not improve every object. Disabling reseeding reduces mean AUC from 86.33 to 69.76. Fixed-candidate controls reveal distribution-specific refit gains and degradation under distribution shift. Overall, the results characterize the accuracy–cost trade-offs and failure boundaries of quality-driven recursive tracking rather than establishing cost-matched autonomous superiority or broad generalization.

1 INTRODUCTION

Recursive 6D object tracking requires deciding not only which pose to output, but when the available hypotheses are inadequate. A visual observation and a temporal prior may disagree, or agree while both are wrong. Ranking them cannot by itself determine whether to select, combine, or replace them. This motivates supervision based on predicted task error rather than relative preference alone.

The central difficulty is that supervisory decisions change the evidence available for later decisions. Selected outputs determine future priors; accepted registrations can restart the observer and change future visual estimates. Quality assessment therefore operates on a *policy-induced candidate distribution*. A predictor accurate on fixed hypotheses need not remain accurate on those generated by its own policy, and correcting the current output need not repair the state that generates the next one.

We formulate *shared task-error supervision for recursive tracking*, linking candidate assessment, hypothesis renewal, and state feedback (Figure 1). A low-capacity predictor makes these interactions inspectable. Our contributions address three connected questions:

(1) Task-error supervision beyond candidate ranking. We assess visual and temporal candidates on a common normalized ADD-S scale, separating absolute eligibility from relative fusion quality. Matched renewal opportunities reveal an average tracking benefit from the combined selection/fusion policy; a separate autonomous control exposes the accuracy–cost trade-off when each policy chooses its own renewal times (Sections 5.2–5.3).

(2) Distribution-specific learning under policy feedback. We refit once on development candidates encountered by the supervisor, then freeze parameters for online use. Same-candidate and cross-distribution controls distinguish predictor changes from changed inputs: prior exposure and sample count alone do not reproduce the observed gain, while cross-distribution reversals reveal adaptation rather than uniform improvement (Section 5.4).

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.

(3) Distinct roles of observer repair and motion history. We separate these two feedback paths through closed-loop interventions. Observer reseeding has a substantial average benefit, whereas resetting history after ordinary registration does not improve mean accuracy. This distinguishes state-update mechanisms that would be obscured by treating reinitialization as a single operation (Section 5.5).

Together, these contributions make recursive quality assessment a joint study of prediction, future candidate generation, and correction persistence. Evidence comes from post-development diagnostics on five inspected sequences; independent generalization and equal-cost autonomous superiority remain open.

2 Related Work

Learning candidate quality. ZePHyR learns to rate 6D pose hypotheses from differences between projected model points and observations, including for unseen objects (Okorn et al., 2021). It is a direct precedent for learned geometric hypothesis assessment. ATOM predicts bounding-box overlap to guide visual tracking (Danelljan et al., 2019); Learning Loss predicts task loss to select samples for annotation (Yoo & Kweon, 2019). Predicting quality to guide an action is therefore not itself new. Our focus is a common task-error target for visual and temporal pose hypotheses whose distribution changes through recursive decisions. Unlike registration ranking alone, this assessment must distinguish a usable pair from one requiring renewal. The gap relative to these quality-learning formulations is to account for how decisions driven by the quality model change the distribution of candidates that the same model must subsequently assess.

Pose generation and recursive tracking. DeepIM refines supplied poses through image matching (Li et al., 2018); se(3)-TrackNet learns local RGB-D pose corrections (Wen et al., 2020). PoseRBPF represents pose uncertainty through particle filtering (Deng et al., 2019), while TP-AE incorporates temporal prediction (Zheng et al., 2022). BundleTrack and BundleSDF retain longer-term information for tracking and reconstruction (Wen & Bekris, 2021; Wen et al., 2023). We retain the backbone rather than propose a new tracker or posterior estimator. FoundationPose includes pose refinement and hypothesis scoring (Wen et al., 2024); its registration-stage scoring is distinct from assessing the observation–prior pair produced by our supervisor. SAM 2 supplies propagated masks (Ravi et al., 2025), and SuperPose already combines FoundationPose and SAM2 with matching machinery (Deng et al., 2024). These capabilities motivate component-matched controls, not novelty through integration. The remaining question is how to unify candidate selection, renewal, and observer-state correction as a recursive supervision problem, in which quality-driven decisions alter subsequent candidate generation.

Calibration and policy-induced data. Isotonic calibration assigns empirical failure probabilities to scores (Niculescu-Mizil & Caruana, 2005), but calibration and discrimination differ (Guo et al., 2017), and reliability can deteriorate under distribution shift (Ovadia et al., 2019). SelectiveNet learns prediction with rejection (Geifman & El-Yaniv, 2019); here an unreliable pair prompts pose acquisition while the tracker retains an operational state. DAgger addresses sequential distribution shift by repeatedly aggregating learner-induced data (Ross et al., 2011). Our narrower procedure replaces an observation-only seed population using one rollout. It learns candidate error rather than expert actions and inherits neither DAgger's aggregation procedure nor its guarantees. These tools and related ideas do not directly address the joint candidate-error assessment problem considered here: a perception supervisor changes both future priors through its selected outputs and future observations through tracker restarts, thereby shaping the candidate distribution on which its quality model operates.

3 Problem Formulation

We track a rigid object with CAD points M, camera intrinsics K, and RGB-D frames $Y_t = (I_t, Z_t)$. A pose $T \in SE(3)$ maps object coordinates to camera coordinates. At each frame, the supervisor receives an observation candidate T_t^o from the visual tracker and a temporal-prior candidate T_t^p, and produces the final output T_t^f. An initial pose and mask are supplied; subsequent ground-truth poses are unavailable online and are used only to construct offline supervision and evaluation labels.

With approximately uniform frame spacing, the temporal prior repeats the previous relative pose transform under the convention of Eq. (1):

$$T_t^p = T_{t-1}^f \big[(T_{t-2}^f)^{-1} T_{t-1}^f \big], \tag{1}$$

where T^f denotes the supervisor's final output. The first two frames initialize the prior from the observation. Because the prior is formed from previous final outputs rather than from a fixed external motion model, current supervisory decisions can directly affect later prior candidates.

For offline supervision, ADD-S error in centimeters and object size are

$$E(T, T^{GT}) = \frac{100}{|M|} \sum_{x \in M} \min_{y \in M} \|T^{GT} x - T y\|_2, \qquad D_M = 100 \| \max_{x \in M} x - \min_{x \in M} x \|_2. \tag{2}$$

Coordinates are in meters; extrema are coordinate-wise. The directed nearest-neighbor convention matches the frozen evaluator: each GT-transformed point queries the predicted cloud. The target is $e = E/D_M$, whereas failure is $E > \tau_E$ with $\tau_E = 1$ cm. Normalization provides a common learning scale, not a scale-invariant policy: routing tolerance remains physical. ADD-S accommodates geometric symmetries but can conceal task-relevant orientation errors.

Let π_q denote the supervisory policy induced by quality model q. Its decisions determine T_t^f, which enters future prior construction through Eq. (1); when a used registration reseeds the visual tracker, the same policy can also alter subsequent observation candidates. We write $\mathcal{D}_{\pi_q}$ for the resulting population of observation–prior candidates encountered under π_q; this is notation for a policy-induced data distribution, not an additional probabilistic model assumed by the method. Consequently, the candidate examples $(\phi(T), e(T))$ available under deployment depend on the quality model through the actions it drives. They are therefore temporally dependent and need not match the observation-only seed population. This learning-under-feedback coupling motivates refitting on policy-encountered candidates, without assuming independent frames, a stationary candidate distribution, or policy convergence.

4 METHOD

The method instantiates the learning-under-feedback formulation in Section 3 with four operations: shared task-error prediction, calibrated eligibility, one-rollout refitting under the induced candidate distribution, and stateful selection or hypothesis renewal. We intentionally use a low-capacity error predictor so that the supervisory mechanisms, rather than scorer capacity, remain the primary object of study.

4.1 A COMMON CANDIDATE-ERROR SCALE

Let q denote the complete quality model and q_θ its regression component. Each candidate is assessed independently against current depth rather than against its competitor. Four CAD–depth statistics are rendered-depth residual x_1 in centimeters, pointwise inconsistency x_2, projected-support deficit x_3, and sparse z-buffer inconsistency x_4:

$$\begin{aligned} \phi(T) &= [x_1/D_M, x_2, x_3, x_4]^\top, & \widetilde{\phi}(T) &= S(\phi(T)), \\ \widehat{e}(T) &= q_\theta(T) = \max\Big(0, w^\top \widetilde{\phi}(T) + b\Big), & \widehat{E}(T) &= D_M \widehat{e}(T). \end{aligned} \tag{3}$$

Here S is a RobustScaler and (w, b) are fitted by Huber regression (Huber, 1964). Observation and prior candidates share the feature definition, scaler, coefficients, and normalized ADD-S target, with no source indicator. Thus both sources are mapped to the same task-error semantics: the model may prefer a prior despite disagreement with the observation, or reject both. $\widehat{E}$ estimates task error, not covariance or a pose posterior. Appendix A gives feature and missing-support details.

4.2 ABSOLUTE ELIGIBILITY AND RELATIVE QUALITY

The same predicted error serves two distinct decisions. On a development calibration partition, isotonic regression fits

$$p(T) = g(\widehat{E}(T)) \approx \Pr(E(T, T^{GT}) > \tau_E \mid \widehat{E}(T)), \qquad c_t^s = \mathbf{1}\{p(T_t^s) \le \tau_p\}, \quad s \in \{o, p\}. \tag{4}$$

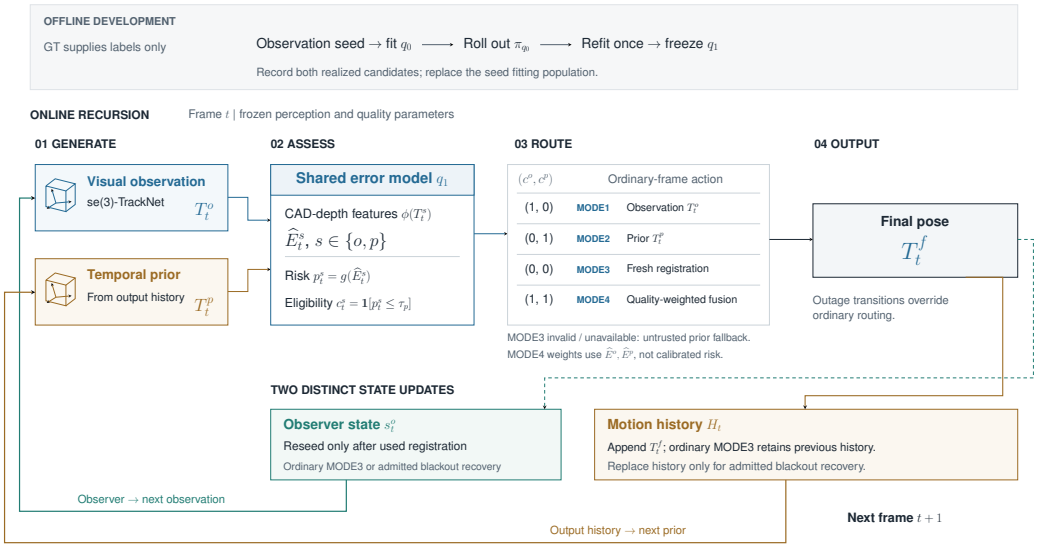

Figure 1: The supervisory loop. A common error model assesses observation and prior candidates, then drives selection, fusion, or fresh registration. Final outputs update motion history; used registrations also reseed the observer. Both updates affect the next candidate pair. In frozen Full, ordinary MODE3 appends the correction without clearing history; only admitted blackout recovery replaces history. GT supplies offline labels, never online actions.

The bounded nondecreasing g minimizes squared probability error. τ_p is selected from $\{0.05, 0.06, \ldots, 0.95\}$ by development balanced accuracy, which gives equal weight to the failure and non-failure classes; the first maximizer is retained. Both are frozen for evaluation, and prior eligibility always uses $p(T_t^p)$.

The *development-calibrated risk score* governs absolute eligibility; continuous $\widehat{E}$ governs relative fusion weights. The fitted g estimates development failure frequencies, not guaranteed probabilities on shifted candidates. Monotonicity adds no ranking information, and isotonic plateaus can discard distinctions retained by $\widehat{E}$. A development-tuned raw-error cutoff controls for routing effects. We distinguish discrimination, probability reliability, and thresholded routing; Table 5 exposes poor prior-risk reliability after development calibration.

4.3 Refitting on recursively encountered candidates

The quality model changes the distribution on which it must later operate. Let $\mathcal{D}_{\text{seed}}$ be the observation-only seed population and $\text{Fit}(\cdot)$ include scaling, regression, calibration, and threshold selection. We perform

$$q_0 = \text{Fit}(\mathcal{D}_{\text{seed}}), \qquad \mathcal{D}_{\pi_{q_0}} = \text{Rollout}(\pi_{q_0}), \qquad q_1 = \text{Fit}(\mathcal{D}_{\pi_{q_0}}). \tag{5}$$

During the development rollout, both candidates are recorded after generation by the actual recursive system: priors use realized output history, and post-correction observations come from the reseeded tracker. GT labels each generated candidate offline and never selects actions. The paired rollout population replaces, rather than augments, the seed population, so q_1 is fitted directly to the source mixture encountered under the development policy; this choice does not imply that aggregation is generally inferior.

Within each development episode, the first 70% of ordered indices fit the scaler and regressor; the final 30% calibrate risk. Paired candidates stay together, and equal-length corruption variants share the boundary. The q_1 parameters and evaluation configuration are then held fixed during subsequent rollouts; observer state and motion history continue to update under the policy. Since q_1 may induce a distribution different from $\mathcal{D}_{\pi_{q_0}}$, one refit is not a fixed-point procedure and carries no convergence

Table 1: Ordinary-frame routing; outage transitions take precedence.

c^o	c^p	Mode	Output
1	0	MODE1	Observation T^o
0	1	MODE2	Prior T^p
0	0	MODE3	Fresh registration; uncertain prior if unavailable or invalid
1	1	MODE4	Relative-quality interpolation

or stability guarantee. We evaluate q_0 and q_1 on identical candidate sets and separately control prior exposure versus policy-induced training data.

4.4 Selection, fusion, and hypothesis renewal

Outside outage transitions, the eligibility pair (c_t^o, c_t^p) defines four actions (Table 1). Eligibility is predicted risk, not known accuracy. MODE1/2 select the sole eligible endpoint. MODE2 has no consecutive-use cap. MODE3 performs *hypothesis renewal* by requesting fresh registration; if no finite, numerically valid transform is returned, the prior remains only as an operational fallback and is still untrusted by the learned criterion. MODE3 has no cooldown and applies no blackout-specific geometric gate, so repeated requests remain possible whenever both candidates fail eligibility.

For MODE4, let α_t be the prior weight:

$$\alpha_t = \mathrm{clip}\left(\frac{\widehat{E}_t^o + \varepsilon}{\widehat{E}_t^o + \widehat{E}_t^p + 2\varepsilon}, 0.25, 0.75\right), \quad \varepsilon = 0.01 \text{ cm}, \tag{6}$$

$$\mathcal{I}(T_a, T_b; \alpha) = \begin{bmatrix} R_a \operatorname{Exp}\big(\alpha \operatorname{Log}(R_a^\top R_b)\big) & (1-\alpha)t_a + \alpha t_b \\ 0 & 1 \end{bmatrix}, \quad T_t^f = \mathcal{I}(T_t^o, T_t^p; \alpha_t). \tag{7}$$

The bounded monotone rule shifts weight toward the lower-predicted-error endpoint; ε regularizes the ratio near zero and clipping prevents either eligible endpoint from vanishing. Rotation is interpolated on $SO(3)$ and translation linearly, not by coupled $SE(3)$ screw motion. These are deterministic task-error weights, not inverse covariances; no Bayes-optimality or endpoint-improvement guarantee is assumed, and no additional disagreement gate is applied.

4.5 Hypothesis acquisition and recursive state transitions

Fresh registration implements hypothesis renewal. SAM2.1 Hiera Small propagates the initial mask causally, and FoundationPose registers current RGB-D with five refinement iterations (Ravi et al., 2025; Wen et al., 2024); registration uses neither current pose candidate, later GT masks, nor future frames. Blackout exits use a dedicated registration with occlusion-aware geometric admission; trigger and gate details are given in Appendix B.

We distinguish the visual observer state s_t^o from motion history H_t. A used MODE3 or admitted blackout registration reseeds the observer,

$$s_t^o \leftarrow \mathrm{Reset}(s_t^o, T_t^{\mathrm{reg}}), \tag{8}$$

while motion history follows

$$H_t = \begin{cases} \{T_t^{\mathrm{reg}}\}, & \text{admitted blackout recovery,} \\ \mathrm{Append}(H_{t-1}, T_t^f), & \text{otherwise.} \end{cases} \tag{9}$$

Thus blackout recovery clears pre-correction history and yields a held prior for one prediction, whereas ordinary MODE3 retains the preceding output and can interpret a correction as motion. Observer reseeding does not remove this prior transient. The complete state machine and an algorithmic summary appear in Appendices A–B.

5 Experiments

We ask how quality-guided decisions perform under matched renewal opportunities, what changes when a simple policy chooses its own requests, how refitting depends on candidate distribution, and which state updates sustain corrections. Each comparison has its own evidence population.

Table 2: Nine-condition cached tracking and closed-loop ablations. AUC/failure rates are sequence-macro percentages. Ordinary requests and their summed timers exclude 25 additional blackout exits per variant; these timers are not end-to-end latency (C.6). Paired differences are percentage points; object-bootstrap intervals describe only three objects.

Method	AUC ↑	Fail@1 ↓	Fail@2 ↓	Ordinary requests	Request time (s)
Full	86.328	18.196	10.080	9,462	22,639.3
Matched simple	79.426	29.216	19.649	9,462	22,610.2
No rollout refit	78.406	31.343	22.614	11,871	28,575.2
No observer reseeding	69.759	40.099	30.376	9,929	22,284.6
MODE3 history reset	86.278	18.109	10.573	8,327	19,717.4
No blackout gate	87.118	17.339	9.271	9,489	22,683.0
Full minus control	Sequence Δ	Object Δ			95% object CI
Matched simple	6.902	5.792			[0.243, 15.826]
No rollout refit	7.923	3.114			[−20.928, 30.584]
No observer reseeding	16.570	16.829			[8.978, 23.382]

5.1 Protocols and evidence populations

We use YCBInEOAT (Wen et al., 2020): four mustard/bleach sequences for quality-model development and five evaluation sequences from three objects (two cracker-box, two sugar-box, one tomato-can trajectory; Appendix C.1). The evaluation sequences are disjoint from quality-model fitting but informed method development. Both tracking studies below are therefore *post-development diagnostics*, not independent generalization tests.

Tracking protocol	Nine-condition cached	Four-condition live
Episodes / frames per method	45 / 44,271	20 / 19,656 scored
Simple control	Matched simple	Autonomous simple
Cost measurements	Ordinary requests / their timers	Actual registrations / live-loop time

The cached protocol uses all nine conditions in C.1; the live protocol uses clean, black10, occ60, and drop60, excludes provided initialization frames, and generates observations/masks online. Compare methods *within* each protocol, not their Full scores across protocols. ADD-S AUC integrates success over 0–10 cm; Fail@1/2 use strict 1/2 cm cutoffs. Sequence-macro scores average conditions within sequence, then the five sequences. Object-balanced scores weight three objects equally; their bootstrap intervals are descriptive given this small sample. Offline refit controls instead reuse four-sequence development candidates (Section 5.4).

5.2 Tracking with matched renewal opportunities

Matched simple inherits Full's ordinary MODE3 request times and otherwise accepts its observation, with common registration/reseeding machinery. This evaluates the combined quality-guided selection/fusion policy conditional on renewal opportunities, not the independent effect of either operation. No-refit runs its own closed loop with q_0 and development-fitted calibration. No-reseed, history-reset, and no-blackout-gate change distinct state/admission rules (Section 5.5).

Full improves AUC by 6.902 points and lowers Fail@1 by 11.020 points with the same 9,462 ordinary requests (Table 2). AUC is higher on four of five sequences (Table 3); the object-balanced gain is 5.792 points, a different estimand. The no-refit object interval spans zero. These are conditional tracking benefits, not evidence about autonomous request timing.

Full exceeds Matched simple in each corruption family, but no-refit exceeds Full on occ60 and drop60 (Table D2); the refit gain is not uniform.

For context, B6 takes the minimum GT error of Full's observation, prior, and actual output on the same completed rollout: 86.683% AUC versus Full's 86.328%. Dominance follows by construction

Table 3: Paired sequence-level AUC differences (percentage points), always *Full minus the control*, computed before rounding. CR/CY: cracker reorient/yalehand0; SY/S1: sugar yalehand0/box1; TY: tomato yalehand0. NR: no reseeding; HR: ordinary MODE3 history reset. Negative entries expose exceptions to the average benefit.

Sequence	Full AUC	Δ Matched simple	Δ No-refit	Δ NR	Δ HR
CR	94.349	+11.736	+14.295	+17.975	−0.881
CY	92.248	+19.916	+46.872	−0.018	+0.712
SY	90.615	−0.057	−0.800	+36.212	−0.049
S1	80.532	+2.673	+0.175	+10.553	+0.506
TY	73.899	+0.243	−20.928	+18.128	−0.036
Sequence macro	86.328	+6.902	+7.923	+16.570	+0.050
Object balanced	84.257	+5.792	+3.114	+16.829	+0.036

Table 4: Four-condition live diagnostic (20 episodes per method). AUC/Fail@1 are sequence-macro percentages. Registrations include blackout recovery; live-loop time excludes setup and offline scoring. Costs are unequal; paired results and failure durations are in D.5.

Method	AUC ↑	Fail@1 ↓	Actual registrations	Live-loop time (h)
Full	79.988	24.437	4,477	4.641
Autonomous simple	58.160	55.135	452	1.852
No-refit	76.146	33.978	5,438	5.317

from including Full's output. This offline diagnostic and the separate two-candidate Local minimum neither change future states nor enter deployable-method rankings or timing claims (C.2).

5.3 Autonomous renewal: accuracy, cost, and failure duration

Autonomous simple requests registration using its own observation's sparse z-buffer inconsistency x_4. Development selection freezes threshold 0.6, one-token capacity, 30-frame refill, and 10-frame cooldown, without retraining the observer or quality models. All four grid candidates fall below the development cost cap; the selected point does not match Full's realized budget. Full, Autonomous simple, and no-refit then run their own observer/mask streams for 60 live evaluation episodes. Matched simple was not rerun under this protocol (D.5).

Full gains 21.828 AUC points over Autonomous simple but executes 9.90 times as many registrations (Table 4). It achieves higher AUC on four sequences but underperforms on tomato: 51.008 versus 81.319 (no-refit: 94.974). Mean per-episode maximum failure-run lengths are 200.8, 171.7, and 197.95 frames, respectively. Lower failure frequency therefore coexists with longer sustained failures: Fail@1 counts all failing frames, whereas the latter statistic averages each episode's longest consecutive run. The comparison characterizes one frozen accuracy–cost operating point; unequal budgets and different selection/fusion behavior prevent attributing the gap to request timing alone.

5.4 Distribution-specific refitting and quality reliability

Refitting improves prediction on policy-encountered candidates but can reverse its advantage on another distribution (Figure 2b). We separate this adaptation question from source-wise risk reliability and thresholded routing.

Same-candidate attribution. Each development-policy population contains 22,770 frame pairs from 36 episodes: 31,842 fit and 13,698 calibration candidates, reused for these diagnostics. "Same candidate" fixes poses and labels across scorers within a population, not across policies. On q_0-policy candidates, Q1 reduces MAE from 8.731 to 6.310 cm and raises risk AUROC from 0.659 to 0.890 relative to Q0 (Table D3); this assessment complements, not replaces, closed-loop no-refit.

Observation-only, duplicated-observation, fixed-policy observation+prior, and rollout-pair controls give MAEs of 3.755, 3.755, 3.619, and 1.475 cm on identical q_1-policy candidates. Prior exposure or

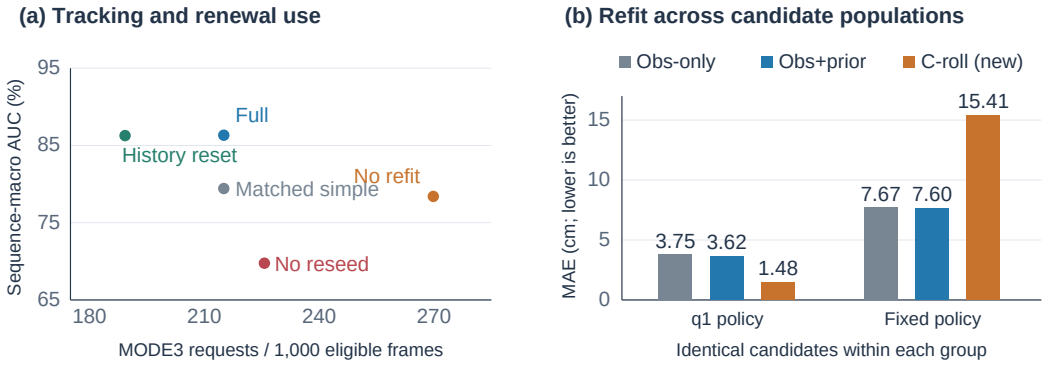

Figure 2: Tracking cost and refit distribution sensitivity. (a) Nine-condition cached protocol; Matched simple inherits Full's ordinary requests, whose frequency is not online throughput. (b) Each group fixes 45,540 development candidates: 31,842 fit and 13,698 calibration indices. C-roll is the offline rollout-refitted control, not frozen Q1 (Table D16); it improves on q_1-policy candidates but degrades on fixed-policy candidates. These diagnostics reuse fitting/calibration indices, not held-out data.

Table 5: Frozen q_1 quality on Full's 45 evaluation episodes. N: candidates; $\bar{y}$: observed failure fraction ($E > 1$ cm). Ref. Brier is $\bar{y}(1-\bar{y})$ for a constant prediction fitted retrospectively to each row, not a deployable baseline. MAE/Brier are lower-is-better; AUROC/pair ordering are higher-is-better. Only pair ordering excludes the 90 GT ties. Candidate-pooled diagnostics do not establish independent generalization or calibration (Appendix C.4).

Source	N	MAE (cm)	AUROC	Brier	$\bar{y}$	Ref. Brier
Observation	44,271	5.088	0.823	0.1823	0.4172	0.2432
Prior	44,271	3.212	0.616	0.2421	0.2329	0.1787
Pooled	88,542	4.150	0.750	0.2122	0.3251	0.2194
Correctly ordered observation–prior pairs						32,547 / 44,181 (73.667%)

sample duplication alone does not reproduce this gain. On fixed-policy candidates, however, C-roll gives 15.407 cm versus C-prior's 7.601 cm: adaptation is distribution-specific. Rollout data bundle selection, fusion, recovery, and reseeding effects rather than isolate one mechanism. Q1 and C-roll are distinct models with risk cutoffs 0.25 and 0.26; C-roll never replaces Q1 in tracking. Appendix D gives all populations, splits, and identities.

Source-wise reliability. Table 5 evaluates Q1 on Full's 88,542 candidates from the five post-development evaluation sequences, pooling candidates rather than averaging sequences. Prior risk has AUROC 0.616 but Brier 0.2421, worse than its retrospective constant reference of 0.1787 (10,311/44,271 failures). Observation risk beats its own reference; pooling obscures this difference. AUROC measures discrimination, Brier overall probability error rather than calibration alone, and routing thresholded decisions. Thus a development-calibrated score need not remain reliable. Tomato drop60 makes the failure concrete: 5.073% AUC despite only eight renewal requests (C.4).

Sharing and routing controls. Shared and source-specific regressors have similar descriptive MAEs (6.310 versus 6.299 cm on q_0 candidates); their AUROC ordering reverses on q_1 candidates. A common target scale therefore does not establish a benefit from parameter sharing. Isotonic and development-tuned raw cutoffs disagree on 0/13,698 calibration and 6/45,540 q_1 candidates (0.013%). This is routing agreement on logged candidates, not statistical equivalence, a separate tracking gain, or evidence of reliable probabilities (D.3).

5.5 State interventions and correction persistence

Observer reseeding and motion-history replacement have different effects. Removing reseeding lowers sequence-macro AUC by 16.570 points and object-balanced AUC by 16.829 points (Table 2), supporting observer-state repair under this protocol. Benefits are heterogeneous: no-reseed is 0.018 points higher on cracker_box_yalehand0. Mean post-MODE3 60-frame error rises from 1.590 to 27.704 cm, but variants trigger at different times and windows overlap; this is descriptive downstream behavior, not a paired effect of one correction.

Adding ordinary MODE3 history reset changes mean AUC by -0.050 points, reduces ordinary requests from 9,462 to 8,327 (12.0%), and increases Fail@2 from 10.080% to 10.573%. It is an ablation, not an improved replacement for Full; fewer requests do not establish a wall-time speedup. Blackout recovery within 60 frames succeeds at 21/25 exits for Full, 19/25 for no-reseed, and 21/25 for history-reset (C.5).

These outcomes use different cohorts: all 25 blackout exits versus all 9,462 ordinary Full requests. The latter yield 8,398 complete and 1,064 end-truncated 60-frame windows (Table C7, including proposal failures). Window means use only complete windows; trajectory AUC retains every frame. They must not be interpreted as all-request recovery probabilities.

Admission can also reject useful corrections: among 25 exits, 22 registrations are accurate, yet the blackout gate rejects five of them while preventing one inaccurate admission. Removing it increases confirmed recovery from 21/25 to 24/25 and AUC by 0.789 points. Together with suppressed renewal on tomato occ60/drop60, this separates errors in requesting a correction from errors in admitting one; neither is resolved by state repair alone.

6 Limitations

Evidence scope. Five inspected sequences represent only three independent objects. Both tracking protocols and the reused development-candidate diagnostics support scoped comparisons, not independent generalization. The autonomous grid covers one low-cost operating region; fresh evaluation and broader development-tuned, cost-matched comparisons remain open.

Attribution. Matched requests constrain renewal opportunities but do not separate selection, fusion, and their downstream state effects. Rollout refitting bundles these effects and specializes to a candidate distribution; one refit is not a convergence procedure. The controls establish no independent tracking gain from sharing or calibration. B6 bounds only the logged candidate/output set, not an optimal recursive trajectory.

Operational reliability. Four geometric features and a linear regressor can be confidently wrong under occlusion or sensing shift. Development calibration does not preserve prior-risk reliability, registration can return inaccurate poses, and the blackout gate rejects accurate recoveries. Live-loop timing is not full application throughput; cached request timers have a narrower scope. These limits preclude universal reliability or efficiency claims.

7 Conclusion

Quality assessment in recursive tracking must be evaluated together with the candidate distribution and state updates it induces. Under matched renewal opportunities, Full improves average accuracy; observer reseeding contributes substantially, whereas ordinary history reset does not improve mean AUC. Autonomous evaluation exposes a different trade-off: higher average accuracy at greater registration cost, with object-specific reversals and longer mean episode-maximum failure runs. Refit gains likewise depend on candidate distribution. The resulting lesson is to assess prediction, state repair, and acquisition cost jointly, while keeping conditional diagnostic evidence distinct from independent generalization.

Reproducibility Statement

Appendices A–D specify models, features, thresholds, and cohorts. The original offline package reproduces 270 historical tracking episodes, quality diagnostics, and corrected B6; its `TABLE_FIGURE_MAP.json` and `PACKAGE_MANIFEST.json` map results and verify files. The separate autonomous study adds 80 development runs (40 verified reused, 40 new) and 60 new evaluation runs; Appendix D.5 specifies its frozen selection, initialization amendment, scoring, and timing. Its logs and integrity records are separate from the historical archive: the original offline package alone does not reproduce this new study. No historical results or frozen models are overwritten. Live reruns require the documented raw data and GPU environment.

Ethics Statement

Predicted low pose risk is not a robot-safety certificate. Deployment requires independent sensing and control safeguards. Harmful corrections, unsuccessful events, and object-specific failures remain part of evaluation.

AI Use Statement

AI tools were used to assist with polishing portions of the manuscript and with software development. AI-assisted code was reviewed and tested by the author. The authors take full responsibility for the final content of this work.

References

Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ATOM: Accurate Tracking by Overlap Maximization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. URL https://arxiv.org/abs/1811.07628.

Xinke Deng, Arsalan Mousavian, Yu Xiang, Fei Xia, Timothy Bretl, and Dieter Fox. PoseRBPF: A Rao-Blackwellized Particle Filter for 6D Object Pose Tracking. In *Proceedings of Robotics: Science and Systems*, volume 15, 2019. doi: 10.15607/RSS.2019.XV.049. URL https://www.roboticsproceedings.org/rss15/p49.html.

Yu Deng, Jiahong Xue, Teng Cao, Yingxing Zhang, Lanxi Wen, and Yiyang Chen. SuperPose: Improved 6D Pose Estimation with Robust Tracking and Mask-Free Initialization. *arXiv preprint arXiv:2409.19986*, 2024. URL https://arxiv.org/abs/2409.19986.

Yonatan Geifman and Ran El-Yaniv. SelectiveNet: A Deep Neural Network with an Integrated Reject Option. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2151–2159. PMLR, 2019. URL https://proceedings.mlr.press/v97/geifman19a.html.

Tilmann Gneiting and Adrian E Raftery. Strictly Proper Scoring Rules, Prediction, and Estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437. URL https://doi.org/10.1198/016214506000001437.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On Calibration of Modern Neural Networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330. PMLR, 2017. URL https://proceedings.mlr.press/v70/guo17a.html.

Peter J. Huber. Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964. doi: 10.1214/aoms/1177703732. URL https://doi.org/10.1214/aoms/1177703732.

Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. DeepIM: Deep Iterative Matching for 6D Pose Estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 683–698, 2018. URL https://openaccess.thecvf.com/content_ECCV_2018/html/Yi_Li_DeepIM_Deep_Iterative_ECCV_2018_paper.html.

Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning (ICML)*, pp. 625–632, 2005. doi: 10.1145/1102351.1102430. URL https://doi.org/10.1145/1102351.1102430.

Brian Okorn, Qiao Gu, Martial Hebert, and David Held. ZePHyR: Zero-shot Pose Hypothesis Rating. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2021. URL https://arxiv.org/abs/2104.13526.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://papers.neurips.cc/paper_files/paper/2019/hash/8558cb408c1d76621371888657d2eb1d-Abstract.html.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollar, and Christoph Feichtenhofer. SAM 2: Segment Anything in Images and Videos. In *International Conference on Learning Representations*, pp. 28085–28128, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/45c1f6a8cbf2da59ebf2c802b4f742cd-Abstract-Conference.html.

Stephane 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. PMLR, 2011. URL https://proceedings.mlr.press/v15/ross11a.html.

Bowen Wen and Kostas Bekris. BundleTrack: 6D Pose Tracking for Novel Objects without Instance or Category-Level 3D Models. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021. URL https://arxiv.org/abs/2108.00516.

Bowen Wen, Chaitanya Mitash, Baozhang Ren, and Kostas E. Bekris. se(3)-TrackNet: Data-driven 6D Pose Tracking by Calibrating Image Residuals in Synthetic Domains. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020. doi: 10.1109/IROS45743.2020.9341314. URL https://arxiv.org/abs/2007.13866.

Bowen Wen, Jonathan Tremblay, Valts Blukis, Stephen Tyree, Thomas Müller, Alex Evans, Dieter Fox, Jan Kautz, and Stan Birchfield. BundleSDF: Neural 6-DoF Tracking and 3D Reconstruction of Unknown Objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 606–617, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Wen_BundleSDF_Neural_6-DoF_Tracking_and_3D_Reconstruction_of_Unknown_Objects_CVPR_2023_paper.html.

Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. FoundationPose: Unified 6D Pose Estimation and Tracking of Novel Objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17868–17879, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Wen_FoundationPose_Unified_6D_Pose_Estimation_and_Tracking_of_Novel_Objects_CVPR_2024_paper.html.

Donggeun Yoo and In So Kweon. Learning Loss for Active Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. URL https://arxiv.org/abs/1905.03677.

Linfang Zheng, Aleš Leonardis, Tze Ho Elden Tse, Nora Horanyi, Hua Chen, Wei Zhang, and Hyung Jin Chang. TP-AE: Temporally Primed 6D Object Pose Tracking with Auto-Encoders. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2022. URL https://lynne-zheng-linfang.github.io/TP-AE.github.io/.

APPENDIX A FEATURES AND LEARNING DETAILS

A.1 CANDIDATE CONDITIONED GEOMETRY

Feature notation and implementation mapping. Equation (3) orders the four features as (x_1, x_2, x_3, x_4); both observation and prior use this same feature vector. The following mapping identifies the frozen implementation without changing its inputs or weights:

Paper	Geometric quantity	Stored field
x_1/D_M	Normalized rendered-depth residual	`x1_norm`
x_2	Point-weighted depth inconsistency	`x2_inlier_error`
x_3	Projected-support deficit	`x4_support_ratio`
x_4	Sparse z-buffer inconsistency	`x5_geometry_inconsistency`

Despite its stored name, `x4_support_ratio` is a deficit, not a support fraction. B3 thresholds paper x_3, whereas Autonomous simple thresholds paper x_4; neither changes the shared predictor's feature set.

Let Z^R denote the mesh-rendered candidate depth and Ω its foreground. The first feature is 100 times the mean absolute depth difference over Ω, in centimeters; zero measured depth is retained in the residual. If at most 20 foreground pixels are rendered, x_1 is set to 20 cm.

For x_2 and x_3, transform each sampled CAD point by the candidate and retain positive-depth points whose rounded projection lies inside the image. Let J be this point-index set and V its subset with positive measured depth. The inconsistency x_2 is one minus the fraction of V with absolute candidate-to-measured depth difference strictly below 0.02 m. It equals one if V is empty. Multiple CAD points can contribute at the same pixel, so this statistic is point-weighted.

The support deficit x_3 is one minus the number of points in J whose measured depth exceeds 0.1 m, divided by $|J| + 10^{-5}$. It is one when the projection is empty. For x_4, retain the nearest candidate depth at every occupied rounded pixel to form a sparse z-buffer. The feature is one minus the fraction of all occupied pixels with measured depth strictly between 0.05 and 5 m and absolute depth discrepancy below 0.02 m. Missing depth remains in this denominator. Fewer than 20 occupied pixels, or no valid measured depth, yields $x_4 = 1$. These fallbacks are shared input conventions, not pose-error labels.

The first residual is divided by the centimeter-valued CAD bounding-box diagonal before regression; the remaining three features are dimensionless. All four are recomputed at the supplied candidate pose using identical conventions for observation and prior. Raw cues are neither ground-truth pose errors nor independent evidence sources. Missing depth, arm occlusion, sparse sampling, and the duplicated projections used by pointwise statistics affect the mapping to ADD-S.

A.2 REGRESSION CALIBRATION AND REFITTING

Huber regression uses $\varepsilon_H = 1.35$, L2 regularization coefficient 10^{-4}, at most 2,000 optimization iterations, and tolerance 10^{-6}. The estimator jointly fits coefficients, intercept, and a positive residual scale; inference clips negative normalized predictions to zero. The RobustScaler uses the default median and interquartile-range normalization. Isotonic calibration is nondecreasing, bounded to [0, 1], and clips out-of-range inputs to endpoint predictions. A calibration split containing only one failure class is rejected rather than assigned a spurious operating threshold.

The first 70%/last 30% split is defined on unique ordered frame indices per episode, with both candidate samples at each index assigned together. Both candidates share the normalized-error target definition and feature ordering. The final fit uses four development sequences and nine conditions; evaluation sequences contribute no regression or calibration examples. The offline pair-label margin is not an online routing criterion: eligibility uses predicted risk, and MODE4 weights use predicted error.

The seed model q_0 includes an observation-trained regressor and its development-fitted calibrator and threshold. A rollout with q_0 collects two examples per frame: the prior from supervisor history and the observation from the active observation stream. Fitting q_1 uses this paired population in place

of the seed population. Subsequent rollouts hold q_1 and the evaluation configuration fixed while observer state and motion history continue to update. No additional refit is performed. Inspected evaluation results informed method development, so exclusion from fitting does not make those sequences untouched test data. Q0 and Q1 denote the corresponding frozen artifacts in D.6.

A same-candidate q_0/q_1 assessment fixes poses and labels before scoring either model. A closed-loop no-refit comparison lets each model produce its own trajectory and request schedule. Calibration is fitted on development data for each model, so thresholds need not match. These comparisons distinguish prediction on fixed inputs from behavior after the policy changes its inputs.

Shared/source-specific, same-candidate q_0/q_1, fixed-policy prior-exposure, and rollout-data controls use the same feature and target definitions and are reported in Appendix D. They are distinct from the closed-loop no-refit runner.

APPENDIX B BLACKOUT ADMISSION AND STATE SEMANTICS

B.1 PIXEL SETS AND GEOMETRIC EVIDENCE

An outage has at most 1% valid full-image depth pixels in $(0.05, 5)$ m. The prior is propagated until depth returns; after at least ten outage frames, the first non-outage frame requests dedicated blackout recovery. Unavailable or rejected proposals leave the prior in place.

Let P contain finite pixels of the candidate's sparse nearest-point CAD z-buffer and V those with finite measured depth strictly between 0.05 and 5 m. Their intersection $C = P \cap V$ is the comparable set. Write δ for measured minus candidate depth and use tolerance $\max(0.01, 0.05 D_M/100)$ meters, where D_M is the centimeter-valued CAD bounding-box diagonal. Foreground occlusion $O \subseteq C$ has δ below minus one tolerance; free-space conflict $F \subseteq C$ has δ above twice the tolerance. The non-occluded comparable set is $W = C \setminus O$. Visible support S intersects W with the propagated mask; inliers $I \subseteq S$ have absolute residual at most one tolerance.

Mask explanation and CAD coverage share an occlusion-removed overlap numerator but have different denominators. The former divides by the occlusion-removed propagated mask; the latter divides by the occlusion-removed projected CAD convex hull. Occlusion removal uses the sparse depth comparison, not a dense mesh visibility model. Every count denominator is clamped below by one. Spatial dispersion counts inlier-occupied cells in a 4×4 grid over the bounding box of the entire sparse CAD projection.

Table B1. Blackout gate thresholds. Every hard condition and all conditions in at least one support path must pass.

Group	Quantity	Requirement
Hard	Finite valid inputs and rigid pose	Required
Hard	Free-space conflict $\lvert F\rvert/\lvert W\rvert$	≤ 0.10
Hard	Visible support $\lvert S\rvert$; positive inliers $\lvert I\rvert$	Both ≥ 50
Hard	Global inliers $\lvert I\rvert/\lvert C\rvert$	≥ 0.10
Normal path	Visible mask explanation; CAD coverage	≥ 0.60; ≥ 0.30
Normal path	Local inliers $\lvert I\rvert/\lvert S\rvert$; spatial cells	≥ 0.60; ≥ 3
Occlusion path	Foreground fraction $\lvert O\rvert/\lvert C\rvert$	≥ 0.10
Occlusion path	Visible mask explanation; CAD coverage	≥ 0.60; ≥ 0.10
Occlusion path	Local inliers $\lvert I\rvert/\lvert S\rvert$; spatial cells	≥ 0.80; ≥ 4
Diagnostic only	Previous-pose jumps and template score	No veto

These geometric gate constants are fixed engineering settings in the frozen configuration, not outputs of isotonic calibration or evidence of an optimal gate. Transform validity requires finite values, a homogeneous last row within absolute tolerance 10^{-6}, and rotation orthogonality and determinant one within 10^{-3}. These checks remain in MODE3 and the no-blackout-gate ablation: they prevent malformed poses, not inaccurate ones. Admission requires every hard condition and every condition of at least one support path.

Potential foreground occlusion is a depth relation, not a robot-arm classifier. A large occluded fraction cannot compensate for insufficient positive support or excessive free-space conflict. Conversely, a large temporal pose jump is not a rejection criterion. The proposed pose is tested against current geometry rather than against a possibly drifted preceding output.

B.2 State transitions after a correction

Table B2. State transition semantics. Reseeding and motion-history reset are different interventions.

Event	Current output	Observer state	Motion history
MODE1 / MODE2 / MODE4	Selected or fused pose	No reseed	Append output
MODE3 valid proposal	Registered pose	Reseed for next frame	Append; retain preceding pose
MODE3 unavailable or invalid	Uncertain prior	No reseed	Append fallback
Blackout admitted	Registered pose	Reseed for next frame	Replace by correction only
Blackout absent or rejected	Prior	No reseed	Append fallback

A one-pose history produces a held prior at the next prediction; it is not a fixed multi-frame freeze. If the next final output equals that pose, the following extrapolation also has zero displacement because its two inputs coincide. New visual evidence can change the output. In contrast, retaining the pre-MODE3 pose admits a correction-induced increment into the very next prior. These effects must be measured from the logged state and candidate histories rather than inferred from the mode name.

Algorithm B1: Shared-quality supervision with one-rollout refitting
Development (once): fit q_0 on $\mathcal{D}_{\text{seed}}$; roll out π_{q_0} on development sequences and log realized observation–prior pairs; label candidate errors offline; replace the seed population by the logged pairs, fit q_1 with the 70/30 regression–calibration split, and freeze it.
Recursive supervision with frozen q_1: (1) obtain T_t^o and construct T_t^p from retained history; (2) during outage use the prior, and at an eligible blackout exit use registration only if admitted by the geometric gate; (3) otherwise evaluate both candidates with Eq. (4) and apply Table 1; (4) reseed the observer after a used registration, clearing motion history only after admitted blackout recovery; (5) log candidates, events, failures, and cost; compute GT pose error offline only.

Appendix C Additional Experimental Results

C.1 Evaluation details

Sequences and protocols. The nine-condition cached study uses mustard_easy_00_02, mustard0, bleach0, and bleach_hard_00_03_chaitanya for quality-model development. Evaluation uses the five sequences in Table D1: two cracker-box trajectories, two sugar-box trajectories, and one tomato-can trajectory. Object disjointness refers to fitting the supervisory quality model, not to training the supplied observation backbone. Each sequence has clean, black10, black10_2, black10_3, black10_4, black10_5, occ40, occ60, and drop60 conditions. D.5 uses a four-condition subset with live perception; its totals and timers must remain separate.

Corruption construction and replay. Clean retains the supplied RGB-D stream. Black10 and its four additional placements remove RGB and depth over ten consecutive frames. Occ40/occ60 apply a rectangular zero-valued RGB/depth occluder over a nominal 40%/60% of the object's image-bounding-box height during a selected temporal window. Drop60 retains RGB and removes depth in 20×20 patches sampled in the object-mask region. The retained generator accumulates patch-mask counts toward a nominal 60% target; overlapping patches can be counted repeatedly, so this is not a measured unique-pixel dropout fraction. Masks used to construct synthetic corruptions are

offline inputs, not subsequent online supervision. All methods replay the same stored corrupted images and frame associations; they do not resample corruptions per policy. Exact placements and realized pixels are determined by those preserved inputs, not by assuming one generator default applies to every sequence. The five blackout placements share an underlying trajectory and are not independent object samples.

Initialization and perception. All methods receive the sequence's supplied initial mask and use the recorded observation trajectory as the initial pose source. In the cached protocol, observations follow that frozen trajectory until a used registration restarts the observer; subsequent observations follow the restarted tracker. SAM2.1 Hiera Small masks are causally precomputed and replayed. FoundationPose uses five refinement iterations and a persistent registration worker. Configurations fix 16 CPU threads, eight I/O workers where applicable, and seed 42. The risk cutoffs 0.25 for q_1 and 0.60 for q_0 are development-fitted. Only admitted blackout recovery clears motion history. Later GT poses are used for scoring, never online decisions. The live protocol instead initializes from the first recorded pose and propagates observation and mask streams online (D.5).

Metrics and averaging. For an episode with errors E_i in centimeters, AUC is computed on 1,000 evenly spaced thresholds in $[0, 10]$ by trapezoidal integration of the empirical success curve, divided by 10 and multiplied by 100. Failure is strict $E_i > 1$ or $E_i > 2$. Overall sequence-macro scores first average nine conditions within each sequence, then average the five sequences. Object-balanced scores additionally average sequences within object before giving the three objects equal weight. Blackout group scores first average the five placements. Consequently, the overall score retains a blackout weight of $5/9$ and is not the unweighted mean of the five displayed condition groups. Paired object bootstrap intervals resample the three object clusters 10,000 times with seed 42, preserving sequence, condition, and method pairing.

C.2 Conventional baselines and same-rollout hindsight diagnostics

Primary controlled per-sequence results are retained in Appendix D (Table D1).

B1 uses the original observation trajectory. B2 interpolates observation and its own recursive prior with fixed weight 0.5. B3 selects observation when its projected-support deficit x_3 (legacy key `x4`) is strictly below 0.4 and otherwise uses its own prior; this is a fixed threshold, not a documented development-tuned optimum. It does not threshold sparse z-buffer inconsistency x_4 (legacy key `x5`). B4 uses a Huber-style weight on the norm of observation–prior pose innovation, with cutoff 0.1; it does not predict task error with a learned regressor. B1–B4 omit Full's relocalization and reseeding; their gaps cannot isolate quality learning. B3 is neither B5 nor either simple control.

Two distinct simple controls. *Matched simple* inherits Full's ordinary renewal schedule and otherwise accepts its own observation, using the common registration/reseeding machinery. The historical tables in Appendices C and D.1–D.4 report this control under its full name. *Autonomous simple* instead computes each ordinary request causally from its own current observation geometry and frozen budget; it never reads Full's schedule. It otherwise uses the same simple action/state adapter, including the shared blackout transitions. Its separate live protocol and results appear only in Table 4 and Appendix D.5. Comparing either control to its within-protocol Full is meaningful; subtracting their scores across protocols does not isolate the effect of autonomous timing.

B6 and Local: same-rollout hindsight. Let $E_t^o = E(T_t^o, T_t^{GT})$, $E_t^p = E(T_t^p, T_t^{GT})$, and $E_t^f = E(T_t^f, T_t^{GT})$ be the logged ADD-S errors in centimeters of Full's observation, prior, and actual final output, respectively. The last includes whichever selected, fused, or registered pose Full actually returned. On this *same realized Full rollout*, we compute

$$E_t^{\mathrm{B6}} = \min\{E_t^o, E_t^p, E_t^f\}, \qquad E_t^{\mathrm{Local}} = \min\{E_t^o, E_t^p\}. \tag{10}$$

B6 is the three-candidate hindsight diagnostic; Local is the explicitly separate two-candidate local oracle. Full's current observation can differ from B1's original observation trajectory after observer reseeding, so the correction uses Full's logged observation rather than the B1 error column. Neither diagnostic runs a new trajectory or feeds its GT selection into future policy, observer, or motion-history updates. No candidates are spliced across different rollouts.

Including the actual policy output guarantees $E_t^{\mathrm{B6}} \leq E_t^f$ at every frame, hence no smaller AUC and no larger failure rates under identical aggregation. This is true *by construction*, not evidence for a superior implementable tracker. Local need not outperform Full because fusion or registration can improve on both endpoints. Neither diagnostic is a global trajectory upper bound; even a separately executed greedy recursive oracle would not automatically be one. We exclude both hindsight diagnostics from deployable-method rankings and latency claims.

Population, validation, and interpretation. Equation (10) is evaluated on matched episode/frame keys for all 45 Full episodes and all 44,271 frames, retaining the units and aggregation in C.1. Reaggregation reproduces B5 and verifies zero frames with $E_t^{\mathrm{B6}} > E_t^f$. The AUC gap $86.683 - 86.328 = 0.355$ points describes this realized candidate/output set, not achievable recursive improvement or proximity to a global optimum. The supplementary manifest links the separately versioned correction, input verification, and regeneration script; superseded B6 logs remain preserved separately and are not used as a verified recursive-oracle trajectory.

Table C1: Recorded baselines and separate offline hindsight diagnostics (%). Seq. AUC and failure rates are sequence-macro; Obj. AUC is object-balanced. The lower block uses GT on Full's rollout and is not a deployable-method ranking. B6 includes Full's output, so its dominance is by construction.

Method	Seq. AUC	Obj. AUC	Fail@1 cm	Fail@2 cm
B1: observation-only	33.622	34.405	70.613	67.397
B2: fixed interpolation	33.610	34.394	70.672	67.396
B3: geometric threshold	34.123	35.119	71.565	68.272
B4: Huber innovation	33.619	34.404	70.647	67.391
B5: Full	86.328	84.257	18.196	10.080
B6: three-candidate hindsight	86.683	84.574	17.148	9.763
Local: two-candidate hindsight	86.123	84.100	20.413	10.086

Table C2: Per-sequence AUC (%). B1–B5 are recorded trackers; B6 (three candidates) and Local (two candidates) are offline diagnostics on Full's rollout, defined by Eq. (10), not additional trackers.

Sequence	B1	B2	B3	B4	B5	B6	Local
cracker_box_reorient	48.032	48.007	45.740	48.023	94.349	94.488	93.557
cracker_box_yalehand0	0.392	0.389	0.359	0.392	92.248	92.752	91.647
sugar_box_yalehand0	29.784	29.766	32.847	29.778	90.615	91.308	90.893
sugar_box1	51.582	51.570	51.569	51.574	80.532	80.842	80.528
tomato_soup_can_yalehand0	38.319	38.317	40.098	38.328	73.899	74.026	73.988

C.3 Condition-level results

Table C3: AUC (%) at each of the nine conditions, averaged over five sequences.

Condition	Obs-only	Matched simple	Full	No-refit	No gate
Clean	66.662	86.959	94.223	77.505	94.222
Black10	16.518	86.731	93.035	79.195	94.256
Black10_2	35.982	87.500	92.147	81.573	94.196
Black10_3	28.510	81.074	90.362	83.454	94.131
Black10_4	36.056	78.856	90.951	78.748	90.957
Black10_5	43.520	79.514	88.491	79.133	88.545
Occ40	49.961	88.254	93.373	86.224	93.373
Occ60	17.835	53.588	58.947	61.795	58.949
Drop60	7.551	72.362	75.428	78.023	75.428

Table C4: AUC (%) at all nine conditions, averaged over five sequences. B1–B5 are recorded trackers. B6 and Local are the separate three- and two-candidate same-rollout hindsight diagnostics of Eq. (10); neither changes Full's state or contributes a deployment claim.

Condition	B1	B2	B3	B4	B5	B6	Local
Clean	66.662	66.632	66.269	66.652	94.223	94.449	94.067
Black10	16.518	16.509	16.838	16.519	93.035	93.283	92.753
Black10_2	35.982	35.971	38.012	35.988	92.147	92.404	91.847
Black10_3	28.510	28.489	33.735	28.505	90.362	90.616	90.254
Black10_4	36.056	36.052	39.099	36.059	90.951	91.118	90.721
Black10_5	43.520	43.499	45.615	43.520	88.491	88.671	88.318
Occ40	49.961	49.949	47.471	49.958	93.373	93.873	93.227
Occ60	17.835	17.830	17.305	17.829	58.947	59.725	59.065
Drop60	7.551	7.557	2.760	7.538	75.428	76.009	74.851

Table C5: Condition-wise failure rates (%). Each entry is Fail@1 cm / Fail@2 cm. Blackout averages five placements before averaging sequences.

Condition	Obs-only	Matched simple	Full	No-refit	No gate
Clean	36.711/32.300	21.216/11.601	10.191/0.904	26.729/20.863	10.171/0.904
Blackout	72.561/69.245	24.403/15.843	13.471/4.999	27.862/19.826	11.932/3.544
Occ40	52.928/49.226	19.075/9.401	9.957/1.873	28.971/16.705	9.957/1.873
Occ60	86.295/85.364	58.456/50.070	47.321/42.199	49.949/44.412	47.361/42.199
Drop60	96.780/93.460	42.185/26.553	28.941/20.746	37.128/22.421	28.901/20.746

The two-candidate Local diagnostic's Fail@1 cm equals the fraction of Full's pairs for which both endpoints exceed 1 cm: 11.767% on clean, 15.532% on blackout, 12.077% on occ40, 49.893% on occ60, and 32.317% on drop60. These fractions identify unavailable accurate endpoints, not whether a subsequently registered or interpolated pose is accurate. Corrected B6 additionally includes that actual output and therefore cannot have larger error than B5 on any frame. The B6–B5 gap is a conditional same-rollout hindsight gap, not pure two-endpoint selection regret or an estimate of what an oracle-driven recursive rollout would achieve.

C.4 Quality reliability, ranking, and routing

Table 5 retains all 44,271 frames from Full's 45 episodes: 44,271 poses per source, 88,542 pooled. These five post-development evaluation sequences differ from Appendix D's four development sequences. Scores weight candidates equally, not sequences. MAE uses centimeter residuals; AUROC ranks risks against $y_i = \mathbf{1}\{E_i > 1\,\mathrm{cm}\}$ with average ranks for ties; Brier is $N^{-1}\sum_i (p_i - y_i)^2$ (Gneiting & Raftery, 2007).

Observation, prior, and pooled failure counts are 18,472/44,271, 10,311/44,271, and 28,783/88,542. Setting a constant prediction to each row's prevalence $\bar{y} = N^{-1}\sum_i y_i$ gives reference Brier $\bar{y}(1-\bar{y})$: 0.2432, 0.1787, and 0.2194. The pooled reference uses pooled prevalence, not averaged source-wise references. These are retrospective, evaluation-label-dependent diagnostics, not deployable baselines. No frames are excluded, and no calibrator or threshold is changed.

Prior Brier is worse than its reference despite above-chance discrimination. Brier combines reliability and resolution, not calibration alone. Calibration concerns agreement between probabilities and outcome frequencies; AUROC concerns ranking; routing concerns thresholded decisions. Neither near-identical routing nor these post-development diagnostics establish independent calibration quality.

Pair ordering compares the signs of $E^o - E^p$ and $\widehat{E}^o - \widehat{E}^p$, excluding 90 exact GT ties. Predicted ties would count as incorrect; none occur among the 44,181 eligible pairs. Temporal dependence and repeated corruptions prevent independent-pair inference; no independent-frame interval is claimed. The supplementary derivation reproduces these diagnostics from preserved inputs without refitting.

Appendix D separates fixed-candidate q_0/q_1, sharing, prior-exposure, rollout-data, and routing controls from the live autonomous study. These comparisons concern different populations and interventions; none supplies fresh held-out evaluation.

On tomato drop60, Full reaches 5.073% AUC versus 95.115% for no-refit, with 8 versus 1,302 ordinary requests. Both candidates exceed 1 cm on 1,273/1,308 frames; median errors are about 37.4 cm versus predictions of 0.457 cm. Low predicted risk can preserve a wrong trajectory and suppress renewal. No-refit's higher accuracy requires substantially more registration, not a cost-free remedy.

C.5 Recovery accounting and sustained tracking

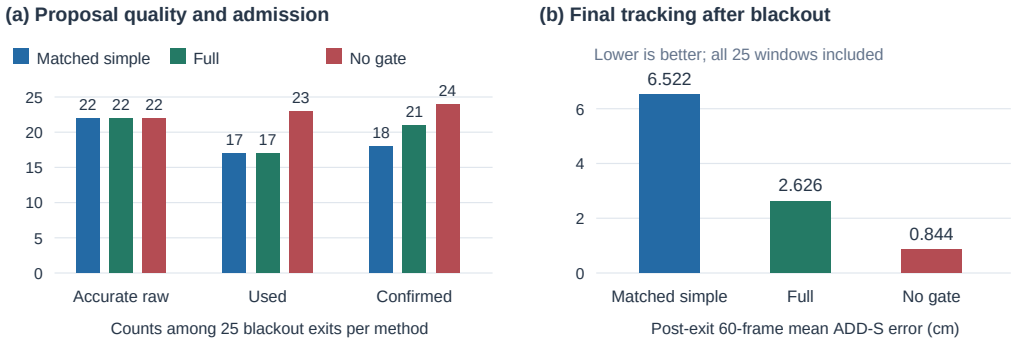

Figure C1: Separating proposal quality, admission, and downstream tracking. Each method has 25 blackout exits. Accurate raw proposals satisfy ADD-S $\leq$ 1 cm; confirmed recovery requires five consecutive accurate final outputs within 60 frames. The bars in (a) are distinct endpoints, not nested subsets: subsequent policy actions can recover after an initial rejection. Panel (b) includes all 25 complete follow-up windows and subsequent interventions. Equal accurate-proposal and admission counts need not yield equal downstream error.

Blackout exits and ordinary MODE3 requests define separate event populations. A valid proposal must be finite and satisfy the numerical rigid-transform checks in Appendix B. Availability uses all triggers as denominator. Raw accuracy is reported against all triggers, including failed generation, and can additionally be conditioned on valid proposals. False rejection divides accurate rejected proposals by all accurate valid blackout proposals; inaccurate-use rate divides inaccurate used proposals by all used proposals. Geometric rejection is inapplicable to ordinary MODE3.

Confirmed blackout recovery is the first run of five final errors at most 1 cm within a 60-frame window including the exit frame. Latency is the offset of its confirming frame and cannot be below four. No confirmation with full follow-up is a horizon failure; insufficient follow-up is sequence-end censoring. All 25 exits in these runs have full follow-up. This distinction is made from actual window lengths rather than a legacy flag that groups unsuccessful events together.

Table C6: Blackout outcomes over the same 25 exit positions per method. NR: no observer reseeding; HR: ordinary MODE3 history reset. Confirmed recovery requires five consecutive accurate outputs within 60 frames. Latency is the confirming-frame offset, conditional on success; failures remain in the separate counts. All post-exit windows include subsequent policy actions.

Outcome	Full	Matched simple	No-refit	NR	HR	No gate
Valid proposals	23	23	23	23	23	23
Accurate valid proposals	22	22	22	22	22	22
Used proposals	17	17	17	17	17	23
Confirmed recovery	21	18	20	19	21	24
Horizon failure	4	7	5	6	4	1
Sequence-end censored	0	0	0	0	0	0
Success-only median latency (frames)	4	4	4	4	4	4
60-frame mean error (cm)	2.626	6.522	3.513	6.721	2.604	0.844
60-frame Fail@1 (%)	22.067	35.867	25.000	44.733	21.267	7.467

The six rejected Full proposals comprise five accurate poses (approximately 0.31–0.45 cm error) and one inaccurate pose (5.96 cm), all rejected for free-space conflict. Four accurate rejections occur on tomato. Its no-gate sequence AUC is 77.841%, compared with 73.899% for Full, accounting for most of the aggregate gate-ablation difference. These cases show why admission precision alone is insufficient: avoiding inaccurate accepted poses can coexist with costly rejection of accurate ones.

Full MODE3 requests registration 9,462 times, uses 8,820 valid poses, and has 8,193 accurate used poses, 627 inaccurate used poses, and 642 requests without a valid proposal. Table C7 accounts for every trigger in each variant: complete windows plus end-truncated windows equal all requests. Each variant uses its own complete-window cohort at every offset, including unsuccessful requests. At a single offset, errors are averaged equally over its complete triggers; for an offset range, each window is averaged over those offsets and then windows are weighted equally. Windows overlap, later interventions are retained, and restricting to complete windows may introduce selection bias. These are descriptive trajectories, not independent event replicates or isolated effects of one correction. Full-trajectory AUC in Table 2 remains the primary state-ablation comparison.

Table C7: MODE3 cohort accounting and downstream ADD-S error (cm). NR: no observer reseeding; HR: ordinary MODE3 history reset. Complete windows contain all offsets 0–59; truncated windows are excluded from every error column. Offset zero is the trigger. These are variant-specific, overlapping cohorts, not paired interventions at identical trigger times.

Variant	Triggers	Complete	Truncated	+1	+2	0–5	0–59
Full	9,462	8,398	1,064	1.343	1.324	1.340	1.590
NR	9,929	8,887	1,042	3.646	4.695	3.683	27.704
HR	8,327	7,386	941	1.412	1.430	1.430	1.663

The observer and history interventions modify different state-update rules and then run their own closed loops. Suppressing reseeding increases average downstream error substantially, with sequence-level exceptions; ordinary history reset has mixed accuracy changes and fewer requests. Neither event-window comparison isolates one correction from later interventions. Disabling blackout-specific recovery would be a distinct experiment and must not be conflated with the completed no-blackout-gate control; no such necessity claim is made here.

C.6 COMPUTATIONAL COST

Table C8: Cached-protocol computation over 45 episodes per variant. Request time sums ordinary MODE3 timers, including unsuccessful requests; cached scope includes initialization and evaluation. Shared SAM preprocessing costs 6,354.1 s once. These overlapping scopes are not full online runtime, and requests are not necessarily registration executions.

Method	Ordinary requests	Requests per 1k	Request time (s)	Cached scope (s)
Full	9,462	215.065	22,639.3	28,889.3
Matched simple	9,462	215.065	22,610.2	28,752.8
No-refit	11,871	269.820	28,575.2	35,161.6
No observer reseeding	9,929	225.680	22,284.6	24,916.0
MODE3 history reset	8,327	189.267	19,717.4	25,000.8
No blackout gate	9,489	215.679	22,683.0	28,871.6

Request rates use 43,996 eligible frames, excluding 250 blackout frames and 25 dedicated exit transitions. All variants additionally request blackout recovery at the 25 exits. Request time sums logged `relocalization_wall_ms`; cached scope sums `episode_scope_wall_ms`, not frame-average timers. The latter includes initialization, frame processing, and evaluation but excludes cold mask preprocessing and the original pre-correction observation pass. SAM preprocessing generates the shared cache once. These component timers overlap and must not be added as disjoint costs. Preserved execution logs identify an NVIDIA GeForce RTX 4090 D with 24 GiB memory; thread/worker settings are in C.1.

State controls were executed separately from the original comparisons, so timing differences can reflect execution conditions as well as requests and observer work. Complete single-pass online wall time is unavailable for these runs: the 12.0% request reduction with history reset is not a measured deployment speedup. Live-loop timing in D.5 has a different scope. B6 and Local are offline GT diagnostics, excluded from tracker rankings, recovery-latency comparisons, and online-runtime claims.

APPENDIX D IDENTIFICATION CONTROLS AND SUPPLEMENTARY PRIMARY RESULTS

D.1 SEQUENCE- AND CORRUPTION-FAMILY SUMMARIES

Tracking tables in D.1 and D.4 use the nine-condition cached protocol; D.2–D.3 assess fixed development candidates, and D.5 uses the four-condition live protocol. Sequence abbreviations are CR: cracker_box_reorient; CY: cracker_box_yalehand0; SY: sugar_box_yalehand0; S1: sugar_box1; TY: tomato_soup_can_yalehand0. Cached-protocol Full, Matched simple, no-refit, no-reseed, and history-reset share 45 episode/frame populations, but not necessarily candidate poses or intervention times.

Table D1: Individual-sequence AUC (%) for the primary controlled comparisons.

Sequence	Obs-only	Matched simple	Full	No-refit	No gate
cracker_box_reorient	48.032	82.613	94.349	80.054	94.353
cracker_box_yalehand0	0.392	72.332	92.248	45.376	92.246
sugar_box_yalehand0	29.784	90.672	90.615	91.415	90.613
sugar_box1	51.582	77.859	80.532	80.358	80.534
tomato_soup_can_yalehand0	38.319	73.656	73.899	94.826	77.841
Sequence macro	33.622	79.426	86.328	78.406	87.118
Object balanced	34.405	78.465	84.257	81.142	85.571

The paired contrasts are presented in main-text Table 3. The no-reseed difference is negative on CY despite a large positive aggregate difference. History reset improves three sequences and worsens two, with little mean AUC change. Full's no-refit contrast is negative on SY and TY; object-balanced uncertainty therefore differs markedly from the sequence-macro point estimate. These contrasts retain all conditions rather than selecting only favorable trajectories.

Table D2: Tracking under each corruption family (sequence-macro AUC, %). Blackout averages black10, black10_2, black10_3, black10_4, and black10_5.

Condition	Obs-only	Matched simple	Full	No-refit	No gate
Clean	66.662	86.959	94.223	77.505	94.222
Blackout	32.117	82.735	90.997	80.421	92.417
Occ40	49.961	88.254	93.373	86.224	93.373
Occ60	17.835	53.588	58.947	61.795	58.949
Drop60	7.551	72.362	75.428	78.023	75.428

D.2 FIXED-CANDIDATE REFIT ATTRIBUTION

These controls use four development sequences, 36 episodes, and 22,770 frame indices. Each paired candidate population contains 45,540 samples: 31,842 at regression-fit indices and 13,698 at calibration indices. The fixed policy always accepts the frozen precomputed observation and never reseeds; its prior is formed from that fixed output history. The q_0- and q_1-policy populations instead contain each policy's realized observation–prior pairs. Candidates and labels are held identical across scorers *within* a population, not across populations.

The first 70% of each episode's ordered indices fit the regressor, and the last 30% fit calibration and select the threshold. Thus neither the pooled nor calibration-partition results are untouched held-out tests; the latter reuses calibration/threshold-selection data, and temporal dependence persists. Cross-population scores diagnose distribution sensitivity on the same development episodes. They do not replace the separate closed-loop no-refit experiment in Table 2.

Table D3: Frozen-model anchors on identical development candidates. Each population pools 45,540 candidates, including regression-fit and calibration indices. Lower MAE and higher AUROC are better; these are not independent test scores.

Candidate population	Scorer	MAE (cm)	Risk AUROC
q_0 policy	q_0	8.731	0.659
q_0 policy	q_1	6.310	0.890
q_1 policy	q_0	3.755	0.743
q_1 policy	q_1	1.476	0.822

Table D4: Complete pooled development matrix. Each cell is MAE (cm) / failure-risk AUROC. Column populations each contain 45,540 candidates. "Rollout refit" denotes the newly fitted control, not the frozen q_1 anchor.

Scorer / training data	Fixed policy	q_0 policy	q_1 policy
Frozen q_0	7.674 / 0.913	8.731 / 0.659	3.755 / 0.743
Frozen q_1	15.409 / 0.895	6.310 / 0.890	1.476 / 0.822
Fixed observation-only	7.674 / 0.913	8.731 / 0.659	3.755 / 0.743
Duplicated observation-only	7.674 / 0.913	8.731 / 0.659	3.755 / 0.743
Fixed observation+prior	7.601 / 0.909	8.605 / 0.644	3.619 / 0.731
Rollout refit (q_0 pairs)	15.407 / 0.897	6.310 / 0.891	1.475 / 0.825

Table D5: Regression-index development matrix: MAE (cm) / failure-risk AUROC on 31,842 candidates per column. Scorers can have fitted these indices under their respective training populations.

Scorer / training data	Fixed policy	q_0 policy	q_1 policy
Frozen q_0	6.467 / 0.859	3.998 / 0.605	3.405 / 0.668
Frozen q_1	12.761 / 0.909	1.742 / 0.895	1.526 / 0.848
Fixed observation-only	6.467 / 0.859	3.998 / 0.605	3.405 / 0.668
Duplicated observation-only	6.467 / 0.859	3.998 / 0.605	3.405 / 0.668
Fixed observation+prior	6.382 / 0.852	3.885 / 0.587	3.299 / 0.653
Rollout refit (q_0 pairs)	12.760 / 0.911	1.742 / 0.896	1.525 / 0.851

Table D6: Calibration-index development matrix: MAE (cm) / failure-risk AUROC on 13,698 candidates per column. These indices are used for calibration and threshold selection, not independent validation.

Scorer / training data	Fixed policy	q_0 policy	q_1 policy
Frozen q_0	10.482 / 0.994	19.732 / 0.755	4.568 / 0.886
Frozen q_1	21.564 / 0.852	16.929 / 0.901	1.362 / 0.790
Fixed observation-only	10.482 / 0.994	19.732 / 0.755	4.568 / 0.886
Duplicated observation-only	10.482 / 0.994	19.732 / 0.755	4.568 / 0.886
Fixed observation+prior	10.436 / 0.995	19.577 / 0.748	4.364 / 0.881
Rollout refit (q_0 pairs)	21.560 / 0.855	16.928 / 0.902	1.359 / 0.794

Table D7: New-control training counts. Duplicated observations match the paired-sample count but add no new candidate geometry. "Fit" and "Cal." count candidate samples, not independent frames.

Training data	Fit	Cal.	Observation	Prior
Fixed observation-only	15,921	6,849	22,770	0
Duplicated observation-only	31,842	13,698	45,540	0
Fixed observation+prior	31,842	13,698	22,770	22,770
Rollout refit (q_0 pairs)	31,842	13,698	22,770	22,770

The observation-only control reproduces frozen q_0 predictions in the recorded anchor check. The new rollout-refit control is close to, but not identical to, frozen q_1: mean absolute prediction difference is 0.003481 cm and maximum difference is 0.031481 cm in that check. We retain both rows rather than substitute the new control for frozen q_1. Rollout collection jointly changes selection, fusion, recovery, and reseeding histories; the data comparison cannot assign the gain to one of those operations. Duplication shows that sample count alone does not explain the observed difference, while the fixed-policy reversal reveals distribution specialization.

D.3 Sharing and calibration controls

Source-specific fitting and scoring. The shared column below uses frozen Q1, not C-roll. The source-specific control fits an observation head and a prior head on the cached q_0-policy development population, with the same features, normalized-error target, and chronological 70/30 split as A.2. Each head has its own RobustScaler, Huber regressor, and isotonic calibrator: 15,921 fit and 6,849 calibration samples per source, totaling 31,842 and 13,698 across heads. Hyperparameters are those in A.2; the failure label is $E > 1$ cm. At evaluation, a sample uses the head corresponding to its source. MAE and risk AUROC pool the resulting predictions over both sources on identical q_0-policy or q_1-policy candidates (45,540 each). These populations reuse development indices, and frozen Q1 is an anchor rather than a newly retrained shared counterpart. The small observed differences do not establish statistical equivalence or an isolated tracking benefit from sharing. D.6 and the package ledger distinguish the fitted control from frozen anchors.

Table D8: Frozen Q1 versus source-specific heads on identical development candidate populations. MAE is in centimeters; AUROC pools source-specific risk scores. These are descriptive prediction comparisons, not independent tracking evaluations.

Population	Shared MAE	Source-specific MAE	Shared AUROC	Source-specific AUROC
q_0 policy	6.310	6.299	0.890	0.885
q_1 policy	1.476	1.479	0.822	0.829

Thresholded routing versus probability reliability. Both rules use Q1's predicted centimeter error. The raw cutoff maximizes balanced accuracy on the 13,698 cached q_0-policy calibration samples, with the smallest cutoff resolving ties; high risk means a strict exceedance. The calibrated rule uses Q1's frozen isotonic map and risk cutoff 0.25. Their agreement is assessed first on those calibration samples and then on all 45,540 q_1-policy candidates, without changing poses or running a new tracker. Agreement of binary decisions neither establishes reliable probabilities nor measures a closed-loop gain. Table 5 separately assesses risk scores on Full's post-development evaluation candidates.

Table D9: Isotonic versus development-tuned raw-error routing on fixed candidates. Balanced accuracy and high-risk rate are fractions; disagreements count candidate decisions.

Population	Routing rule	Balanced acc.	High-risk rate	Route disagreement
Development cal.	Isotonic ($\tau_p = 0.25$)	0.865964	0.284713	0 / 13,698
Development cal.	Raw error (1.070 cm)	0.865964	0.284713	0 / 13,698
q_1 policy	Isotonic	0.685006	0.204150	6 / 45,540
q_1 policy	Raw error	0.685034	0.204282	6 / 45,540

D.4 Recursive-state interventions

These are separate closed-loop executions with frozen quality and perception parameters. No-reseed suppresses the observer restart after used ordinary and blackout registrations but retains the rest of Full's history semantics. History-reset retains observer reseeding and additionally replaces motion history after a used ordinary MODE3 registration. Neither variant alters the published frozen Full result.

Table D10: Closed-loop state interventions. Post-MODE3 is mean ADD-S over offsets 0–59 on each variant's complete-window cohort.

Variant	Seq. AUC	Obj. AUC	Fail@1	Ordinary requests	Post-MODE3
Full	86.328	84.257	18.196	9,462	1.590
No observer reseeding	69.759	67.427	40.099	9,929	27.704
MODE3 history reset	86.278	84.221	18.109	8,327	1.663

Table D11: State interventions by sequence. AUC/failure rates are means over nine conditions (%); ordinary MODE3 requests and request time are totals. NR: no observer reseeding; HR: ordinary MODE3 history reset. Abbreviations are in D.1; timing scope is in C.6.

Sequence	Variant	AUC	Fail@1	Fail@2	Ordinary requests	Request time (s)
CR	Full	94.349	6.193	0.622	613	1,668.9
	NR	76.374	23.378	19.526	482	1,288.5
	HR	95.230	1.630	0.296	423	1,205.2
CY	Full	92.248	10.567	2.805	7,392	18,002.4
	NR	92.266	9.956	2.805	7,517	17,238.9
	HR	91.536	15.172	5.175	6,504	15,668.7
SY	Full	90.615	21.224	5.766	524	1,500.2
	NR	54.403	68.108	51.275	686	1,633.1
	HR	90.664	21.047	5.644	511	1,488.8
S1	Full	80.532	27.649	16.452	921	1,414.5
	NR	69.980	46.098	30.663	923	1,343.3
	HR	80.026	27.478	17.065	874	1,290.4
TY	Full	73.899	25.348	24.754	12	53.4
	NR	55.771	52.956	47.613	321	780.8
	HR	73.935	25.221	24.686	15	64.3

Table D12: All nine conditions for the state interventions. Each cell is AUC / Fail@1 cm / Fail@2 cm (%), averaged equally over five sequences. No conditions or unsuccessful frames are discarded.

Condition	Full	No reseeding	History reset
Clean	94.223 / 10.191 / 0.904	86.961 / 16.143 / 10.253	94.317 / 8.848 / 1.468
Black10	93.035 / 10.288 / 2.463	66.352 / 50.102 / 33.740	93.068 / 10.563 / 2.866
Black10_2	92.147 / 12.063 / 3.387	66.033 / 51.439 / 36.968	92.248 / 11.154 / 3.906
Black10_3	90.362 / 15.167 / 4.895	70.738 / 43.547 / 29.450	90.606 / 13.788 / 5.090
Black10_4	90.951 / 14.769 / 7.122	77.669 / 33.692 / 24.930	90.959 / 15.668 / 7.710
Black10_5	88.491 / 15.069 / 7.129	72.751 / 33.755 / 27.123	88.645 / 14.135 / 7.627
Occ40	93.373 / 9.957 / 1.873	91.413 / 19.932 / 7.244	92.979 / 13.242 / 1.989
Occ60	58.947 / 47.321 / 42.199	44.385 / 59.875 / 56.497	57.670 / 48.238 / 44.080
Drop60	75.428 / 28.941 / 20.746	51.528 / 52.408 / 47.183	76.010 / 27.349 / 20.423

D.5 Autonomous simple: frozen live diagnostic

Question and populations. Matched simple evaluates the combined quality-guided policy under Full's ordinary renewal schedule; Autonomous simple evaluates one nonlearned policy with its own causal schedule and budget. Neither comparison separates request timing from all downstream selection/state effects. Autonomous simple is not B3: B3 selects an existing candidate using support deficit x_3, whereas Autonomous simple requests registration using z-buffer inconsistency x_4. The four development and five inspected evaluation sequences in C.1 are restricted to clean, black10, occ60, and drop60. Development comprises 16 episodes for Full and each of four simple configurations (80 runs); evaluation comprises 20 episodes each for Full/Q1, Autonomous simple, and no-

refit/Q0 (60 new runs). Each evaluation method retains 19,676 frames and scores 19,656, excluding only each supplied initialization frame. All planned episodes, including failures, are retained with matching frame keys across methods.

Causal geometric trigger and budget. At every frame, Autonomous simple measures $x_4(T_t^o)$ on its own live observation. On ordinary frames after initialization, it requests registration if $x_4(T_t^o) > \gamma$, at least 10 frames have elapsed since its last ordinary request, and a token is available. The bucket starts with one token, has capacity one, and replenishes by $1/R$ per elapsed frame (including outage frames). Every ordinary request consumes the token, even if registration fails or is not executed. Ordinary requests are disabled during blackout and its dedicated exit transition; the shared blackout-recovery request is exempt from this bucket but included in actual-call cost. Otherwise the control accepts its observation; used registrations reseed the observer, with the same fallback, admission, and history semantics as Full. The action adapter receives a current-frame Boolean, never a loaded Full request schedule. No learned risk, GT, future observation, or competitor's decisions enter the simple trigger.

Development selection and freezing. The predeclared grid is $\gamma \in \{0.4, 0.6\}$ and $R \in \{30, 60\}$ frames, with cooldown fixed at 10. Selection maximizes development sequence-macro AUC among candidates whose sequence-macro actual calls per 1,000 scored frames do not exceed development Full (219.190). Ties prefer fewer calls, then the stable configuration identifier. Blackout calls count in this selection cost. All four candidates are feasible (Table D13); $\gamma = 0.6, R = 30$ is frozen before evaluation. The cap is a development feasibility constraint, not a requirement that Full and Autonomous simple expend equal budgets at evaluation. Indeed the grid underuses this cap; it does not establish a best geometric policy or a matched-cost frontier. Quality models, observer parameters, and registration weights are reused without training or evaluation-set tuning.

Table D13: All four autonomous development candidates; equal-sequence averages over four conditions. Cost counts actual ordinary and blackout registrations. No evaluated candidate is omitted.

γ	Refill R	AUC (%)	Actual registrations / 1,000	Decision
0.4	30	80.321	34.187	Retained
0.4	60	76.097	17.982	Retained
0.6	30	80.570	26.189	Selected
0.6	60	75.597	13.649	Retained

Live execution and initialization. All methods use identical input-frame lists, frozen observer weights, causal SAM2, and registration/reseeding code. The initial pose is the first entry in the provided observation trajectory; the initial mask is the supplied sequence mask. After initialization, each policy generates its own observations and propagates its own mask online, rather than replaying precomputed streams. No later GT pose enters online computation.

Numerical-protocol amendment. An initialization-validity failure during development prompted a uniform amendment before evaluation: valid seeds remain unchanged; invalid positive-determinant rotations whose maximum singular-value deviation is at most 0.01 are projected by SVD to the nearest $SO(3)$ matrix, preserving translation. Other invalid seeds abort. This affects all four conditions of mustard0 in development and sugar_box_yalehand0 in evaluation. Raw GT annotations are never projected; online validity checks, weights, thresholds, and the grid remain unchanged. Forty bleach development runs meeting the same protocol and input/output-integrity checks are reused once; 40 development and all 60 evaluation runs are newly executed. The failed launch is preserved, not counted as an additional completed episode. The manifest records this amendment and reuse, which preclude describing the study as an untouched pre-registered test.

Scoring and failure duration. The directed ADD-S convention in Eq. (2) and the 1,000-threshold AUC in C.1 are retained. Accuracy and failure-rate averages weight the four conditions equally within each sequence and then the five sequences equally; they are not frame-pooled or object-balanced. Table D14 reports paired differences before rounding. Failure spans are consecutive scored frames with $E > 1$ cm. Table D15 reports the within-episode mean and longest span,

each averaged over the four conditions; an episode with no failure contributes zero. Boundary-truncated spans remain included and are flagged in the logs. Durations are observed frame counts, not estimates of eventual recovery time, and no seconds conversion is inferred without verified frame timing.

Table D14: Paired autonomous-study AUC (%) over four conditions per sequence. Differences are Full minus the indicated control. The negative tomato results are retained. This is not the historical matched-simple cohort.

Sequence	Full	Autonomous simple	No-refit	Δ Autonomous simple	Δ No-refit
CR	94.307	63.042	86.335	31.265	7.972
CY	92.054	22.059	35.915	69.995	56.139
SY	87.996	69.049	89.591	18.947	−1.595
S1	74.575	55.332	73.917	19.243	0.658
TY	51.008	81.319	94.974	−30.310	−43.965
Sequence macro	79.988	58.160	76.146	21.828	3.842

Table D15: Autonomous-study failure and cost by sequence. Fail@1 is percent; mean/max. failure-run lengths are averaged episode statistics in frames, not pooled runs or worst-case guarantees. Actual registrations and live-loop seconds are totals over four episodes.

Seq.	Method	Fail@1	Mean run	Mean max. run	Actual reg.	Live loop (s)
CR	Full	6.350	2.94	9.50	396	1515.1
	Autonomous simple	55.147	22.12	62.50	33	618.5
	No-refit	38.035	4.60	23.50	152	914.7
CY	Full	9.540	3.33	32.75	3,289	9562.7
	Autonomous simple	95.626	29.16	52.75	176	1885.7
	No-refit	79.317	38.38	658.25	375	2410.5
SY	Full	24.850	22.48	129.50	425	2232.3
	Autonomous simple	47.502	57.12	265.00	85	1328.8
	No-refit	18.956	7.44	82.75	624	2693.5
S1	Full	32.671	67.10	202.75	357	1966.9
	Autonomous simple	50.276	211.12	395.25	33	1131.9
	No-refit	28.780	199.52	203.25	460	2189.4
TY	Full	48.776	198.94	629.50	10	1432.3
	Autonomous simple	27.123	24.05	83.00	125	1703.8
	No-refit	4.801	3.54	22.00	3,827	10933.2

Requests, calls, and timing. The saved evaluation environment identifies one NVIDIA GeForce RTX 4090 D (24 GiB; driver 595.71.05) on Linux, with the 16-thread CPU budget in C.1. The retained environment record captures GPU/software versions but not the CPU model; these are within-study timings, not hardware-normalized benchmarks. Live input loading is serial, unlike the optional cached-protocol prefetch workers.

Full, Autonomous simple, and no-refit issue 4,576, 451, and 5,433 ordinary requests, respectively, plus five blackout-recovery requests each. Actual registration executions are 4,477, 452, and 5,438; a requested renewal can terminate before execution, so these counters are not interchangeable. The live-loop totals are 16,709.3, 6,668.8, and 19,141.3 s. They include serial RGB/depth I/O, causal mask propagation, observation updates, feature/scoring/routing work, registration, reseeding/history, lazy worker loading, and lightweight progress logging. Initial setup adds 152.3, 61.3, and 156.6 s, respectively; GT scoring, final result serialization, and worker shutdown are outside the loop. There is one timed execution per method/episode, not repeated performance benchmarking. These measurements do not convert historical cached timers into online FPS.

Interpretation and remaining evidence. Full's average AUC advantage accompanies greater cost, not equal-cost superiority. The mean per-episode mean failure-run length is 58.96 frames versus 68.71 for Autonomous simple (no-refit: 50.70), whereas the mean per-episode maximum is 200.8 versus 171.7 (no-refit: 197.95). Lower total failure frequency need not shorten the longest sustained failures. On tomato, Full executes 10 registrations and reaches 51.008 AUC; Autonomous simple executes 125 and reaches 81.319, while no-refit reaches 94.974 with 3,827. This is consistent with the earlier risk-overconfidence failure, but does not identify request timing as the sole cause: selection, fusion, and evolving candidates also differ. The negative result is retained without evaluation-set retuning. The completed comparison establishes one post-development accuracy–cost trade-off beyond an inherited schedule. Broader development-tuned, cost-matched policies are needed to assess autonomous acquisition advantages, and fresh independent evaluation is needed for independent-generalization claims; neither conclusion follows from these diagnostics alone.

D.6 Model identities and reproducible evidence

Table D16 assigns stable identifiers to the frozen anchors and newly fitted controls. Near-identical predictions do not imply identical models. The fitting/calibration populations, sample counts, and development-fitted risk cutoffs remain explicit; none of these cutoffs is selected on the evaluation sequences. Exact artifact identities and locations are supplied in the supplementary model ledger rather than encoded in the model names.

Table D16: Stable model identities and development fitting populations. "Fit" and "Cal." count candidate samples. Q0/Q1 are frozen tracking anchors; C-* models are offline attribution controls, not replacement tracking models. All populations use the four development sequences; Q1 and C-roll use different recorded rollout-fitting artifacts.

ID	Model / fit population	Fit	Cal.	τ_p
Q0	Frozen q_0, observation seed	15,921	6,849	0.60
Q1	Frozen q_1, original rollout	31,842	13,698	0.25
C-obs	Fixed observation-only	15,921	6,849	0.60
C-dup	Duplicated observations	31,842	13,698	0.60
C-prior	Fixed observation+prior	31,842	13,698	0.47
C-roll	New refit on cached q_0 pairs	31,842	13,698	0.26

Q0 is the observation-seed anchor; Q1 is the original rollout-refitted tracking model. C-obs and C-dup isolate observation exposure and duplicated sample count; C-prior adds fixed-policy prior examples; C-roll refits on cached q_0-policy pairs. A cached rollout is not assumed to reproduce the original Q1 fitting trajectory bitwise. The distinct Q1 and C-roll cutoffs and prediction differences are therefore retained, rather than relabeling C-roll as Q1. The original training revision was not recorded, so present-day code cannot retrospectively certify that lineage; preserved model and source artifacts delimit what can be reproduced.

The historical supplementary table/figure map links its reported results to these model/run identifiers, configurations, source logs, and reproduction scripts using package-relative paths. File-integrity records, input verification, and regeneration commands reside in that package; identifying development records are retained in a separate private provenance archive rather than the anonymized supplement. The original evidence and the B6-hindsight correction remain separate, and corrected three-candidate B6 is never merged with two-candidate Local. These offline derivations are distinct from the new live rollouts in D.5 and do not supply untouched evaluation or a separate recursive oracle. The autonomous-study evidence additionally records its development selection, frozen configuration, numerical amendment, per-episode frame metrics, timing, and integrity checks; it must not be conflated with the earlier 270-episode archive.