LECOMPASS: TASK-LEVEL EXECUTION THROUGH STATE-INDEXED INTERFACES

Anonymous authors
Paper under double-blind review

ABSTRACT

Action-conditioned joint-embedding predictive architectures (JEPAs) offer a foundation for predictive decision-making by modeling the consequences of actions in representation space. Yet evaluating local reachability in latent space does not by itself provide a task-level execution interface: intermediate targets must be generated, local inference has finite budgets, and execution requires spatial feedback and persistent interaction state. We introduce LeCompass, a state-indexed execution framework built around a frozen LeWorldModel (LeWM). A proposer supplies intermediate latent targets, while CompassFlow searches the source space of a learned trajectory prior to obtain effective action candidates under limited predictive-evaluation budgets. Patch-token readouts provide spatial feedback, while observation-based memory supports phase transitions and bounded recovery. State-indexed targets support cross-episode context–goal composition. On 1,000 official TEST full-episode Cube trials, the complete LeCompass execution system achieves 86.9% native 4-cm reach. Separate matched analyses show that predictive ranking raises success from 4.7% to 44.9% on TEST PushT and from 33.8% to 77.7% on TEST TwoRoom, whereas Cube retains 86.4% native reach without predictive ranking. These results demonstrate task-level execution with task-dependent roles for latent target proposal, predictive search, and observation-grounded control.

1 INTRODUCTION

A central promise of world models is to support deliberation by predicting action consequences (Ha & Schmidhuber, 2018; Hafner et al., 2019). Joint-embedding predictive architectures (JEPAs) learn predictive latent spaces without reconstructing pixels (LeCun, 2022; Assran et al., 2023; Bardes et al., 2024). LeWorldModel (LeWM) uses an action-conditioned JEPA to rank CEM-sampled action sequences by the distance between their predicted outcome latents and the target latent (Maes et al., 2026b), enabling local planning over roughly 25 frames.

Local reachability alone does not provide task-level execution: the controller must choose intermediate targets without oracle waypoints, allocate limited predictive compute, and determine actual progress from spatial feedback and accumulated evidence.

We introduce LeCompass, a task-level execution framework based on *state-indexed outer execution with time-indexed local imagination*. The frozen LeWM predicts over a local temporal horizon, while the outer loop re-indexes task progress from observations. This separates task progress from prediction time, trajectory generation from predictive evaluation, and predicted reachability from realized execution state.

A proposer amortizes directed demonstration transitions into latent references conditioned on observed history and final goal, recomputing them after each executed segment. CompassFlow uses a state-conditioned, target-independent trajectory prior: references guide action selection through frozen-LeWM scoring rather than direct action supervision. Patch-token readouts provide execution geometry, while observation-based memory accumulates task-specific temporal evidence for action realization, phase transitions, and bounded recovery.

State-indexed proposal raises cross-episode composition success from 2.9% to 63.7% on PushT and from 39.1% to 60.5% on TwoRoom. At 192 predictive evaluations per plan, CompassFlow Source-CEM reaches $52.5 \pm 2.5\%$ and $77.3 \pm 1.0\%$, respectively, versus 24.4% and 37.2% for

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.

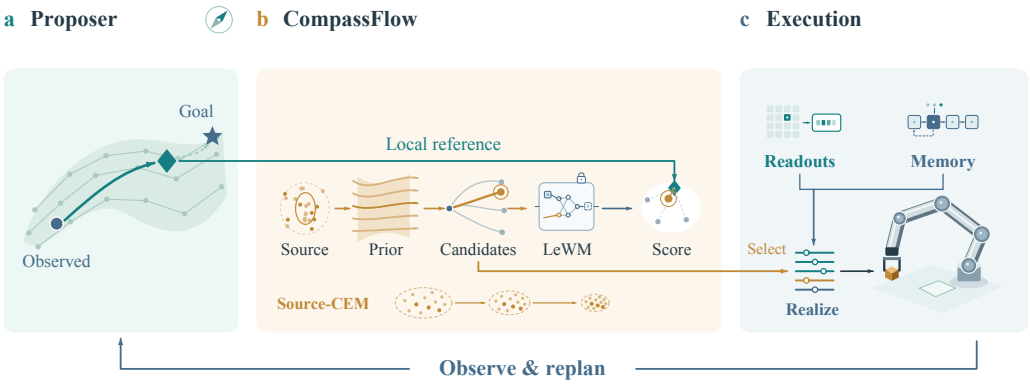

Figure 1: **Three interfaces from local prediction to task-level execution.** (a) Observed history and the final goal yield a local latent reference. (b) Target-independent CompassFlow generates action candidates; frozen LeWM scores them against the reference, and Source-CEM refines sources over three 64-sample rounds. (c) Cube uses patch-token readouts and observation-based memory for action realization, phase transitions, and bounded recovery.

action-space CEM. On the 1,000-episode official Cube panel, the complete system achieves 86.9% native reach; bounded recovery raises semantic completion under induced contact loss from 5.8% to 80.4%. In a separate matched analysis, execution without predictive ranking achieves $86.4 \pm 1.0\%$ native reach on Cube. A height-informed scripted controller reaches 92.5% semantic completion with the shared execution stack and additional proprioceptive height, showing that the stack supports high completion with simple action generation.

We contribute:

- state-indexed latent references for cross-episode goal composition and reorientation from observed progress;
- CompassFlow, combining a target-independent trajectory prior with frozen-LeWM scoring and budgeted source-space search; and
- an observation-grounded Cube execution interface with patch-token readouts, temporal state, and bounded recovery.

2 Related Work

Predictive world models and latent planning. Latent world models and JEPAs support representation-space planning (Ha & Schmidhuber, 2018; Hafner et al., 2019; Hansen et al., 2024; LeCun, 2022; Assran et al., 2023; Bardes et al., 2024). PLDM and DINO-WM use learned or pretrained visual representations (Sobal et al., 2025; Zhou et al., 2025), while V-JEPA 2-AC and LeWM add action conditioning for visual control (Assran et al., 2025; Maes et al., 2026b). LeCompass embeds a frozen action-conditioned JEPA in a task-level execution system.

Hierarchical targets and temporal abstraction. Visual MPC optimizes actions toward visual goals (Finn & Levine, 2017); hierarchical methods add subgoals, latent plans, or high-level goal policies (Nair & Finn, 2020; Lynch et al., 2019; Hafner et al., 2022; Park et al., 2023). SuSIE generates image subgoals (Black et al., 2023); $\pi_{0.7}$ conditions a generalist action model on subtask instructions and generated multi-view subgoal images (Physical Intelligence et al., 2026); and HWM learns multi-timescale predictors (Zhang et al., 2026). LeCompass instead uses state-indexed latent references with one frozen short-horizon predictor, requiring neither test-time visual waypoints nor a predictor hierarchy.

Generative trajectory priors and inference-time search. Diffusion and flow models generate plans or actions (Janner et al., 2022; Chi et al., 2023; Lipman et al., 2023; Zhang et al., 2025);

inference-time methods select or refine noise candidates (Ma et al., 2025), including through diffusion mutations (Yang et al., 2024). LeFlow generates goal-conditioned latent paths, decodes them through inverse dynamics, and verifies them with a frozen world model (Huang et al., 2026). CompassFlow instead searches source coordinates of a target-independent action-trajectory prior, scoring decoded candidates with frozen LeWM rather than searching raw action space.

Action generation and observation-grounded execution. Vision-language-action and world-action models map multimodal context or predicted futures to actions (Kim et al., 2024; Black et al., 2025; Ye et al., 2026; Wang et al., 2026a;b). INTACT maps latent intent to a largely search-free JEPA action distribution, with optional local CEM (Sun et al., 2026). Execution monitoring and behavior trees organize behavior through persistent state, applicability conditions, and recovery (Pettersson, 2005; Ögren & Sprague, 2022). LeCompass combines this execution-side organization with test-time candidate generation and frozen-JEPA predictive evaluation.

3 Problem Formulation

We consider goal-conditioned visual control with observation o_t, action a_t, and goal observation o_γ. A frozen JEPA encoder provides global planning latents and spatial patch tokens; its action-conditioned predictor estimates candidate outcomes over a supported local horizon H:

$$(z_t, p_t) = E_\theta(o_t), \quad (z_\gamma, p_\gamma) = E_\theta(o_\gamma), \quad \hat{z}_{t+H} = F_\theta(z_t, a_{t:t+H-1}). \tag{1}$$

At replanning round j, the controller evaluates at most B candidates, executes a prefix of length $h_j \leq H_j$, and replans from the next observation. Success is defined by a domain-specific task predicate.

State-indexed task progress. Observed latent history $\mathbf{z}_j^{\text{hist}}$, final-goal latent z_γ, and local span κ_j determine a new reference at each replanning round:

$$z_j^g = P_\psi(\mathbf{z}_j^{\text{hist}}, z_\gamma, \kappa_j), \qquad H_j = b\kappa_j, \tag{2}$$

Here b counts raw actions per latent step. Unlike an elapsed-time waypoint schedule, the proposer re-indexes its reference from observed progress.

Target-independent generation and target-dependent evaluation. A target-independent trajectory prior maps source coordinates ξ to action candidates, scored against z_j^g using frozen-model predictions:

$$\mathbf{a}_j(\xi) = \mathcal{T}_\phi(\xi; \mathbf{z}_j^{\text{hist}}, \kappa_j), \qquad \xi_j^\star = \arg\min_{\xi \in \Xi_{B,j}} \left\| F_\theta(z_{t_j}, \mathbf{a}_j(\xi)) - z_j^g \right\|_2^2. \tag{3}$$

The evaluated source set $\Xi_{B,j}$ contains at most B coordinates, each requiring one predictive evaluation.

Observation-grounded action realization. Predicted proximity to a reference does not establish realized task progress. Spatial readouts and persistent memory therefore govern action realization:

$$r_t = R_\rho(p_t, p_\gamma), \quad m_t = U_m(m_{t-1}, r_t, e_t), \quad \tilde{a}_t = \mathcal{X}(a_t^{\text{plan}}; r_t, m_t). \tag{4}$$

Readouts r_t expose execution geometry, using p_γ only for goal-relative outputs. Memory m_t accumulates observation-derived evidence e_t, such as contact; $\mathcal{X}$ combines planned actions, readouts, and memory into the environment action $\tilde{a}_t$.

The proposer P_ψ, prior $\mathcal{T}_\phi$, and readouts R_ρ are learned with the JEPA frozen; the memory update U_m and action realization $\mathcal{X}$ are non-learned (Sec. 4.3).

4 LeCompass

Figure 2 summarizes the closed loop of target proposal, trajectory search, and action realization.

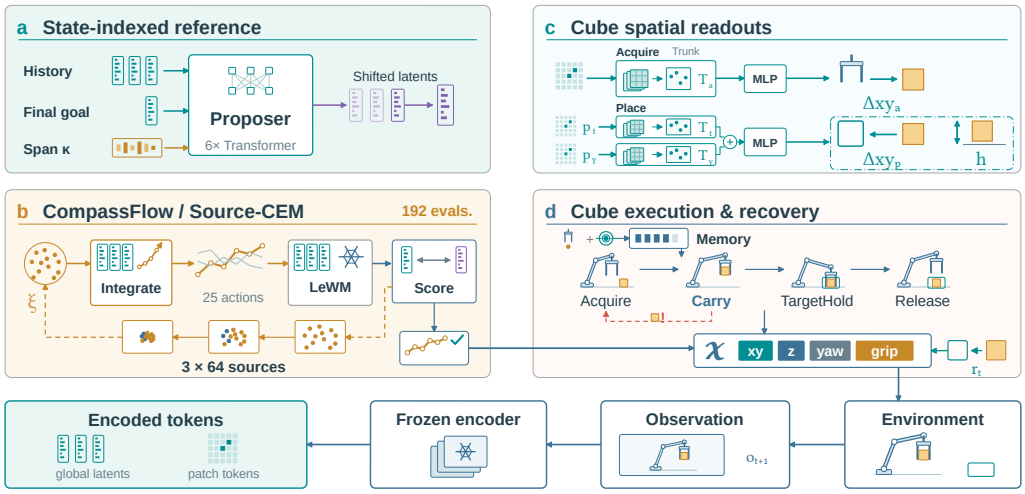

Figure 2: **LeCompass planning and execution interfaces.** **(a)** History, final goal, and span κ condition the proposer; its latest inverse-normalized prediction supplies the scoring reference. **(b)** CompassFlow decodes 25-action candidates with eight Euler steps. Source-CEM evaluates three 64-source rounds, using frozen LeWM to score predicted outcomes. The reference enters scoring, not the trajectory prior. **(c)** Cube acquisition reads current patch tokens; placement also uses cached final-goal tokens to estimate displacement and block height. **(d)** Observation-based memory confirms phases and permits bounded recovery from contact loss in CARRY. The placement controller assigns action channels, after which grip-floor and release actuators may override the gripper command. Environment observations update the encoded history and patch tokens.

4.1 STATE-INDEXED LATENT PROPOSAL

A frozen short-horizon predictor evaluates local reachability but cannot direct successive local predictions. The proposer amortizes directed transitions in demonstration latents into history- and goal-conditioned local references. It maps the latest three normalized latents, a later-goal latent, and span κ to a shifted three-latent sequence. The conditioning goal z_g is sampled from the context trajectory during training and set to z_γ at evaluation. With latent dimension $D = 192$, normalization ν, latent stride Δ, and $H = b\kappa$, training minimizes

$$\mathcal{L}_{\text{prop}} = \mathbb{E}\left[\frac{1}{3D}\sum_{q\in\{-2,-1,0\}}\left\|[P_\psi^{\text{seq}}(\bar{\mathbf{z}}_t^{\text{hist}}, \nu(z_g), \kappa)]_q - \nu(z_{t+q\Delta+H})\right\|_2^2\right]. \tag{5}$$

The inverse-normalized latest prediction supplies z_j^g, a synthesized directional reference for candidate scoring rather than a retrieved demonstration frame or a physical waypoint that must be reached exactly.

Recomputing references from observed history supports cross-episode context–goal composition and reorientation after deviations. The proposer and trajectory prior are trained across supported spans; all reported deployments use $\kappa = 5$. Appendix B.1 specifies sampling, normalization, and architecture.

4.2 COMPASSFLOW: SOURCE-SPACE TRAJECTORY SEARCH

A synthesized reference may lack a paired expert trajectory. CompassFlow learns a state-conditioned, target-independent prior over demonstrated action chunks; references guide test-time selection through frozen-LeWM scoring.

For normalized expert action chunk $\bar{\mathbf{a}}$, source noise $\epsilon \sim \mathcal{N}(0, I)$, and interpolation time $\tau \sim \mathcal{U}[0, 1]$, define $x_\tau = (1 - \tau)\epsilon + \tau\bar{\mathbf{a}}$. The rectified-flow velocity field minimizes

$$\mathcal{L}_{\text{flow}} = \mathbb{E}\left[\underset{M_\kappa=1}{\text{mean}} \left(v_\phi(x_\tau, \tau; \bar{\mathbf{z}}^{\text{hist}}, \kappa) - (\bar{\mathbf{a}} - \epsilon)\right)^2\right], \tag{6}$$

The mask M_κ marks the valid $b\kappa$ action positions. The field conditions on observed history and span, not z_j^g. Integrating v_ϕ defines $\mathcal{T}_\phi$, implemented with eight Euler steps.

At inference, Source-CEM (Rubinstein, 1999) scores candidates using Eq. 3, allocating 192 predictive evaluations across three 64-candidate rounds. It fits a diagonal Gaussian to low-cost source coordinates while retaining base-distribution samples, then selects the lowest-cost trajectory across all rounds. Appendix B.2 specifies masking and action normalization; Appendix B.3 specifies source-bank construction, source constraints, ranked IID sampling, and the complete Source-CEM update.

4.3 Spatial Readout and Execution Memory

Patch-token readouts estimate current geometry; observation-supported memory accumulates confirmed execution conditions across replanning. Together they govern action-channel assignments and overrides, actuator constraints, phase transitions, and prefix invalidation when supporting conditions fail. Memory is non-learned; its structure and task-specific parameters are determined from development rollouts and fixed before evaluation.

Cube instantiation. Acquisition uses current patch tokens; placement additionally uses cached final-goal tokens:

$$r_t^{\text{acq}} = R_{\rho_a}(p_t), \qquad r_t^{\text{place}} = R_{\rho_p}(p_t, p_\gamma).$$

These goal tokens are independent of the intermediate ranking reference. Readouts are adapted on development data with the encoder frozen, then fixed for evaluation.

After trajectory selection, readouts guide lateral control; remaining planner commands follow phase-specific pass-through and actuator constraints. During pre-pick, vertical and yaw commands pass through, while the observation-based grip floor may override the gripper command. Accumulated contact and placement evidence confirms phases, maintains gripper constraints during carry, and gates release. When recovery is enabled, detecting sustained contact loss in CARRY prompts the controller to discard the unexecuted prefix and initiate bounded reacquisition, preserving observed latent history and outer-loop task progress.

The readout targets, evidence predicates, temporal thresholds, channel assignments, and recovery budgets are task-specific. The appendices specify the complete Cube execution program and readout adaptation.

5 Experimental Setup

Environments and protocols. We evaluate PushT planar object pushing, collision-aware TwoRoom navigation, and full-episode Cube pick-and-place (Maes et al., 2026a;b). T1 tests nominal execution; T2 pairs contexts and goals from different episodes; T3 perturbs execution while the controller observes and replans. Cube T3 forces gripper opening after stable carry. Task-level, primary search, and ranking-attribution comparisons use official TEST panels. Learned LeCompass components use non-TEST training data; LeWM remains frozen.

Comparisons. Target-interface comparisons use the original action-space CEM (9,000 predictive evaluations per plan) with direct final-goal targets, elapsed-time Fixed-10/25 demonstration waypoints, or state-indexed proposals. Search comparisons evaluate action-space CEM, ranked flow sampling, and CompassFlow Source-CEM at 192 predictive evaluations per plan; separate attribution studies vary the scoring target or predictive ranking. The PushT action-space-versus-Source-CEM comparison also changes the candidate prior, proposer refresh cadence, and warm-start behavior, so it is not a search-space-only ablation. Cube Recovery ON/OFF changes only the bounded recovery bundle, retaining the same readouts, monitoring, and perturbation rule.

Metrics and reporting. PushT uses joint object-and-tool success; TwoRoom requires target-region entry; Cube reports native 4-cm reach and post-release semantic completion. Paired comparisons use exact McNemar tests. Appendix A specifies panels, schedules, seeds, success predicates, baseline definitions, attribution-arm protocols, and implementation details.

6 RESULTS

6.1 STATE-INDEXED TARGET COMPOSITION

Table 1(a) tests cross-episode context–goal composition without an episode-specific waypoint schedule. Despite training on within-trajectory pairs, state-indexed proposal raises success from 2.9% to 63.7% on PushT and from 39.1% to 60.5% on TwoRoom. With a test-time demonstration schedule (Table 1(b)), it is comparable to Fixed-25 on PushT T1 and stronger under T3 perturbation (29.0% versus 20.2%; exact McNemar $p = 7.9 \times 10^{-4}$), consistent with re-indexing targets from observed progress. Fixed-25 remains stronger on TwoRoom T1 and T3; Sec. 7.1 examines the related frozen-latent geometry.

Table 1: **State-indexed target composition.** Success (%) on official TEST panels with 9,000 predictive evaluations per plan. (a) Cross-episode T2 without an episode-specific waypoint schedule. (b) T1/T3 with test-time demonstration schedules; Fixed-k advances through demonstration waypoints every k raw steps. – denotes an unrun arm. State-indexed PushT T1 is mean $\pm$ sample SD over three proposer checkpoints; all other entries are single fixed runs (Appendix A).

(a) No episode-specific waypoint schedule: cross-episode T2

Domain	Context–goal pairs	Direct final goal	State-indexed target
PushT	1,000	2.9	**63.7**
TwoRoom	1,000	39.1	**60.5**

(b) Episode-specific demonstration schedule available

Domain	Protocol	N	Direct	Fixed-10	Fixed-25	State-indexed
PushT	T1	500	5.2	57.0	**59.2**	58.1±1.0
	T3	500	4.2	–	20.2	**29.0**
TwoRoom	T1	500	14.0	29.0	**40.6**	31.2
	T3	500	–	–	**32.6**	28.4

6.2 FINITE-BUDGET SOURCE-SPACE INFERENCE

At 192 predictive evaluations per plan, CompassFlow Source-CEM reaches $52.5 \pm 2.5\%$ success on PushT and $77.3 \pm 1.0\%$ on TwoRoom, versus 24.4% and 37.2% for action-space CEM (Table 2). These are configuration comparisons: the PushT arms also differ in candidate prior, search space, proposer refresh cadence, and warm-start behavior. Within Source-CEM, replacing the final-goal scoring target with the proposed reference improves mean success by 38.6 percentage points on PushT and 15.3 points on TwoRoom. Matched comparisons with predictively ranked IID sampling show that adaptive source refinement adds 6.2–9.4 points across the three PushT checkpoints but no mean gain on TwoRoom (Appendix F).

6.3 FULL-EPISODE CUBE EXECUTION AND RECOVERY

On the common 1,000-episode official TEST panel, LeCompass achieves 86.9% native reach and 86.0% semantic completion, versus 3.7% and 4.3% native reach for Direct and Fixed-25 (Table 3(a)). These baselines lack the same goal-aware execution program, so the gap does not isolate predictive planning. Under the matched T3 forced-opening intervention, enabling the bounded recovery bundle raises full-panel semantic completion from 5.8% to 80.4%; both arms retain execution monitoring (Table 3(b)).

In a separate shared-stack comparison, a scripted rule reaches 92.5% semantic completion without predictive scoring but with additional proprioceptive end-effector height. This is neither a same-input

Table 2: **Finite-budget local inference.** Success (%) at 192 predictive evaluations per plan on official TEST PushT (500 episodes) and strict TEST TwoRoom (500 windows). h is the nominal maximum execution prefix in raw actions. Source-CEM reports mean ± sample SD over three field checkpoints; action-space CEM uses single fixed runs.

Domain	Search	Target	h	Success (%)
PushT	Action-space CEM	Proposed latent	5	24.4
	CompassFlow Source-CEM	Proposed latent	5	**52.5±2.5**
	CompassFlow Source-CEM	Final-goal latent	5	13.9±1.2
TwoRoom	Action-space CEM	Proposed latent	25	37.2
	CompassFlow Source-CEM	Proposed latent	25	**77.3±1.0**
	CompassFlow Source-CEM	Final-goal latent	25	62.0±2.0

prior ablation nor a controlled comparison with the headline configuration (Appendix G.3). Figure 3 illustrates nominal execution and bounded recovery.

Table 3: **Full-episode Cube execution and bounded recovery.** (a) T1 native 4-cm reach, semantic completion, and predictive budget on 1,000 official TEST episodes. Native reach includes frame zero; completion requires completed release and post-third-opening containment. B/h is nominal predictive evaluations per environment step; – denotes not evaluated. (b) Matched T3 Recovery ON/OFF under the same perturbation, readouts, and monitoring; only bounded recovery changes. Further diagnostics are in Appendix G.

(a) T1: full-episode execution and predictive budget

Method	Native reach (%) ↑	Semantic completion (%) ↑	Evals/ plan ↓	Nominal B/h ↓
Direct final-goal CEM	3.7	–	9,000	360.0
Fixed-25 demonstration-oracle CEM	4.3	–	9,000	360.0
LeCompass	86.9	86.0	192	38.4

(b) T3: full-panel bounded-recovery intervention

Outcome	Recovery ON	Recovery OFF
Native reach	837/1000 (83.7%)	397/1000 (39.7%)
Semantic completion	**804/1000 (80.4%)**	58/1000 (5.8%)

7 Analysis

7.1 How Does Target Geometry Differ Across Domains?

Let E be median proposer error, G the 95th-percentile global-pair distance, and A the median one-latent-step transition distance. The ratio $E/(G-A)$ is 0.137 on PushT, 0.227 on Cube, and 2.924 on TwoRoom. TwoRoom's narrow local-to-global separation relative to proposer error is consistent with Fixed-25's advantage when an episode-specific demonstration schedule is available (Table 1). This checkpoint-conditioned distance diagnostic does not measure dynamical reachability; Appendix E reports the full scales and uncertainty intervals.

7.2 What Does Predictive Search Contribute?

On official TEST panels, matched proposed-target IID best-of-192 selection improves success over unranked execution by 40.2 ± 0.2 percentage points on PushT and 43.9 ± 2.7 points on cadence-25 TwoRoom across three field checkpoints (Fig. 4(a)). Unranked arms retain the encoder and trajectory prior but use no predictive evaluations. In a separate matched Cube IID suite, unranked execution reaches $86.4 \pm 1.0\%$ native reach. Final-goal ranking changes reach by -1.9 ± 1.1 points, while replacing the final-goal target with the proposed reference adds 2.8 ± 0.7 points. The net $+0.9 \pm 0.9$-point effect over unranked execution remains unresolved (Fig. 4(b); Appendix F).

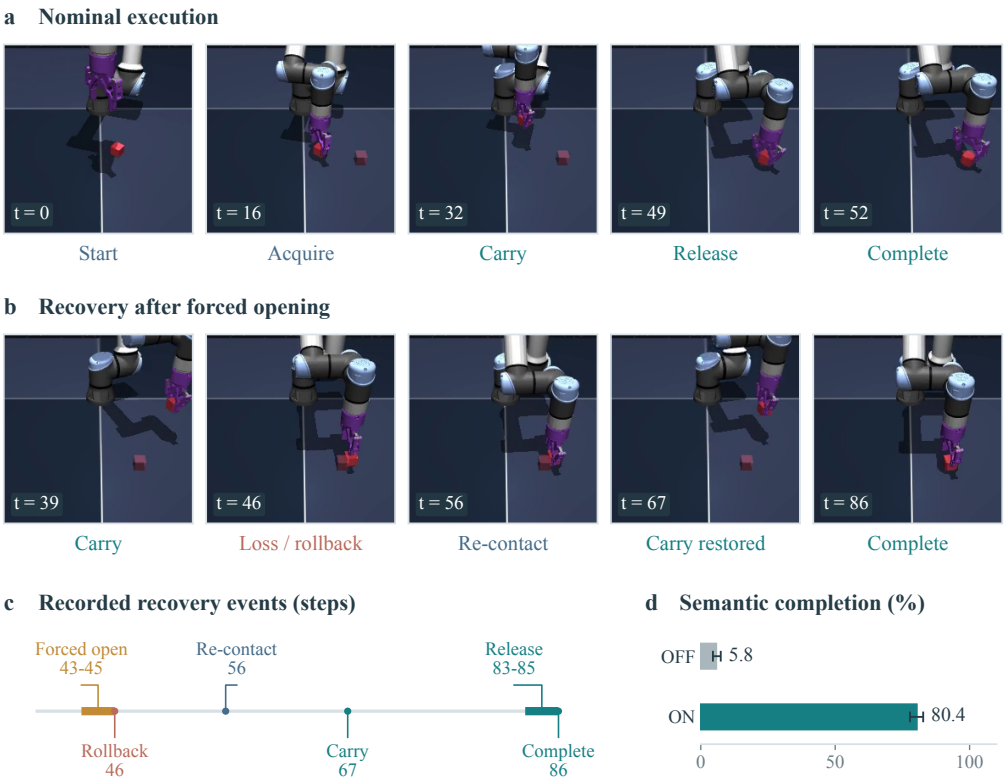

Figure 3: **Observation-grounded Cube execution and bounded recovery.** (a,b) Nominal execution and forced-opening recovery in different official TEST episodes; frames are shown at nonuniform steps. (c) Forced opening at $t = 43$–45 leads to rollback at $t = 46$, discarding three unexecuted actions. (d) Matched full-panel T3 semantic completion (1,000 episodes): 5.8% OFF versus 80.4% ON; whiskers show 95% Wilson intervals.

7.3 Why Does Execution Need Observation-Grounded State?

The tested projected-CLS MLP decodes end-effector XY well but block-relative XY poorly (Appendix D.4); the spatially corresponding patch token ranks first in all seven non-null swaps. With the encoder and predictor frozen, readout adaptation reduces median acquisition and placement errors from 63.0 to 9.2 mm and from 190.5 to 11.9 mm on shared controller-induced states (Fig. 4(c)). Separate validation rollouts show corresponding pick and placement gains (Appendix D). Readouts provide current geometry, while memory confirms execution conditions; when enabled, sustained contact loss in CARRY prompts the controller to initiate bounded recovery (Sec. 6.3). LeWM scores five-channel candidates before execution-time channel assignments; the effect of this scoring–execution difference on ranking is not isolated (Appendix G).

8 Limitations

Evaluation covers three simulation benchmarks and released LeWM checkpoints, not physical robots. The complete observation-grounded execution interface is instantiated only on Cube; its task-specific readouts, evidence and timing rules, channel assignments, and recovery budgets may require new development rollouts and adaptation to perception shift, calibration error, latency, contact noise, and safety constraints. Cube outcomes, including those of a scripted rule with additional proprioceptive height, do not establish that predictive scoring or the learned prior is necessary for completion. Finite-budget search comparisons use a single budget of 192 predictive evaluations per plan, and the PushT configuration comparison is not a search-space-only ablation. Cube T3 tests the complete

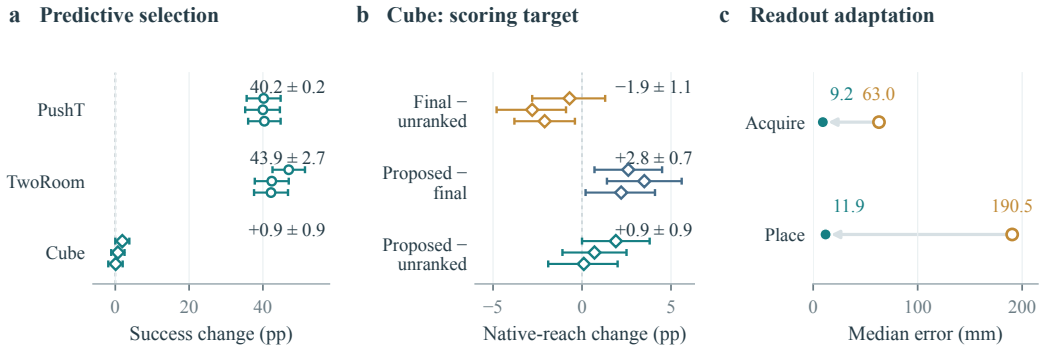

Figure 4: **Predictive selection and spatial readout evidence.** **(a)** Proposed-target IID best-of-192 selection versus matched unranked execution on official TEST PushT, strict TEST TwoRoom, and Cube. Points represent three field checkpoints for PushT/TwoRoom or three source seeds for Cube; whiskers are within-repeat paired 95% intervals, and labels give mean ± sample SD. **(b)** Scoring-target contrasts in the same Cube IID suite, separate from the headline run; the net proposed-minus-unranked effect remains unresolved. **(c)** Initial (open amber) and adapted (filled teal) median XY readout errors on identical controller-induced frames: 5,338 for acquisition and 5,580 for placement. Separate closed-loop validation improves confirmed picks from $47/113$ to $93/113$ and, with acquisition fixed, placement native reach from $11/93$ to $92/93$ on the same 93 confirmed-pick episodes.

recovery bundle under one event-triggered forced-opening disturbance, not its components, natural drops, repeated failures, or other disturbance types.

9 Conclusion

LeCompass organizes frozen-JEPA local prediction into task-level execution through state-indexed latent references, budgeted source-space search, and observation-grounded control. State-indexed targets support cross-episode composition, while predictive selection substantially helps PushT and TwoRoom. Cube retains high success without predictive ranking, while readout adaptation and bounded recovery improve execution outcomes. These findings demonstrate complementary, task-dependent roles for predictive search and observation-grounded execution around a frozen world model in the evaluated settings.

AI Use Statement

We used AI tools to assist with manuscript drafting and language refinement, and improvements to the organization and consistency of the main text, figures, tables, and appendices. These tools also supported literature retrieval and discovery, including identifying relevant work and suggesting search directions. For research ideation, we used them to discuss methodological choices and refine experimental design. The authors reviewed and revised the AI-assisted content, made the final methodological and editorial decisions, and take full responsibility for the text, references, methods, experiments, analyses, and claims presented in this work.

References

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. URL https://arxiv.org/abs/2301.08243.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-JEPA

2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025. URL https://arxiv.org/abs/2506.09985.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*, 2024. URL https://arxiv.org/abs/2404.08471.

Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pretrained image-editing diffusion models. *arXiv preprint arXiv:2310.10639*, 2023. URL https://arxiv.org/abs/2310.10639.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. In *Robotics: Science and Systems*, 2025. URL https://arxiv.org/abs/2410.24164.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems*, 2023. URL https://arxiv.org/abs/2303.04137.

Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *IEEE International Conference on Robotics and Automation*, 2017. URL https://arxiv.org/abs/1610.00696.

David Ha and Juergen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018. URL https://arxiv.org/abs/1803.10122.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, 2019. URL https://arxiv.org/abs/1811.04551.

Danijar Hafner, Kuang-Huei Lee, Ian Fischer, and Pieter Abbeel. Deep hierarchical planning from pixels. *arXiv preprint arXiv:2206.04114*, 2022. URL https://arxiv.org/abs/2206.04114.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Oxh5CstDJU.

Hsiang-Wei Huang, Jianxu Shangguan, Junbin Lu, and Jenq-Neng Hwang. LeFlow: Generative latent flow planning for world models. *arXiv preprint arXiv:2608.24855*, 2026. URL https://arxiv.org/abs/2608.24855.

Michael Janner, Yilun Du, Joshua B. Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning*, 2022. URL https://arxiv.org/abs/2205.09991.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. URL https://arxiv.org/abs/2406.09246.

Yann LeCun. A path towards autonomous machine intelligence. OpenReview, 2022. URL https://openreview.net/forum?id=BZ5a1r-kVsf.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.02747.

Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. *arXiv preprint arXiv:1903.01973*, 2019. URL https://arxiv.org/abs/1903.01973.

Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, and Saining Xie. Scaling inference time compute for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2523–2534, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Ma_Scaling_Inference_Time_Compute_for_Diffusion_Models_CVPR_2025_paper.html.

Lucas Maes, Quentin Le Lidec, Luiz Facury, Nassim Massaudi, Ayush Chaurasia, Francesco Capuano, Richard Gao, Taj Gillin, Dan Haramati, Damien Scieur, Yann LeCun, and Randall Balestriero. stable-worldmodel: A platform for reproducible world modeling research and evaluation, 2026a. URL https://arxiv.org/abs/2605.21800.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. LeWorldModel: Stable end-to-end joint-embedding predictive architecture from pixels. *arXiv preprint arXiv:2603.19312*, 2026b. URL https://arxiv.org/abs/2603.19312.

Suraj Nair and Chelsea Finn. Hierarchical foresight: Self-supervised learning of long-horizon tasks via visual subgoal generation. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1909.05829.

Petter Ögren and Christopher I. Sprague. Behavior trees in robot control systems. *Annual Review of Control, Robotics, and Autonomous Systems*, 5:81–107, 2022. doi: 10.1146/annurev-control-042920-095314.

Seohong Park, Dibya Ghosh, Benjamin Eysenbach, and Sergey Levine. HIQL: Offline goal-conditioned RL with latent states as actions. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2307.11949.

Ola Pettersson. Execution monitoring in robotics: A survey. *Robotics and Autonomous Systems*, 53 (2):73–88, 2005. doi: 10.1016/j.robot.2005.09.004.

Physical Intelligence, Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, Thomas Charbonnier, Vedant Choudhary, Foster Collins, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Maitrayee Dhaka, Jared DiCarlo, Danny Driess, Michael Equi, Adnan Esmail, Yunhao Fang, Chelsea Finn, et al. $\pi_{0.7}$: A steerable generalist robotic foundation model with emergent capabilities. *arXiv preprint arXiv:2604.15483*, 2026. URL https://arxiv.org/abs/2604.15483.

Reuven Y. Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and Computing in Applied Probability*, 1(2):127–190, 1999. doi: 10.1023/A:1010091220143.

Vlad Sobal, Wancong Zhang, Kyunghyun Cho, Randall Balestriero, Tim G. J. Rudner, and Yann LeCun. Learning from reward-free offline data: A case for planning with latent dynamics models. *arXiv preprint arXiv:2502.14819*, 2025. URL https://arxiv.org/abs/2502.14819.

Junhan Sun, Hao Zhao, and Guofeng Zhang. INTACT: Isomorphic intent-to-action learning for search-free world models. *arXiv preprint arXiv:2607.26056*, 2026. URL https://arxiv.org/abs/2607.26056.

Rui Wang, Yue Zhang, Jiehong Lin, Kuncheng Luo, Jianan Wang, Zhongrui Wang, and Xiaojuan Qi. When to trust imagination: Adaptive action execution for world action models. *arXiv preprint arXiv:2605.06222*, 2026a. URL https://arxiv.org/abs/2605.06222.

Siyin Wang et al. World action models: The next frontier in embodied AI. *arXiv preprint arXiv:2605.12090*, 2026b. URL https://arxiv.org/abs/2605.12090.

Brian Yang, Huangyuan Su, Nikolaos Gkanatsios, Tsung-Wei Ke, Ayush Jain, Jeff Schneider, and Katerina Fragkiadaki. Diffusion-ES: Gradient-free planning with diffusion for autonomous and instruction-guided driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15342–15353, 2024. doi: 10.1109/CVPR52733.2024.01453.

Seonghyeon Ye et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026. URL https://arxiv.org/abs/2602.15922.

Qinglun Zhang, Zhen Liu, Haoqiang Fan, Guanghui Liu, Bing Zeng, and Shuaicheng Liu. FlowPolicy: Enabling fast and robust 3D flow-based policy via consistency flow matching for robot manipulation. In *AAAI Conference on Artificial Intelligence*, 2025. URL https://arxiv.org/abs/2412.04987.

Wancong Zhang, Basile Terver, Artem Zholus, Soham Chitnis, Harsh Sutaria, Mido Assran, Randall Balestriero, Amir Bar, Adrien Bardes, Yann LeCun, and Nicolas Ballas. Hierarchical planning with latent world models. *arXiv preprint arXiv:2604.03208*, 2026. URL https://arxiv.org/abs/2604.03208.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. *arXiv preprint arXiv:2411.04983*, 2025. URL https://arxiv.org/abs/2411.04983.

A Experimental Protocols

A.1 Evaluation panels and data separation

Tables 4 and 5 distinguish target-interface, search, and attribution panels. T1 tests nominal execution; T2 pairs contexts and goals from different episodes; T3 perturbs realized execution. Proposer training uses within-trajectory context–goal pairs, whereas T2 uses different TEST episodes excluded from downstream training. PushT T2 uses 1,000 unique episodes in each role, with disjoint role sets and each episode used once.

Table 4: **Target-interface and full-episode Cube protocols.** N counts episodes for PushT/Cube, strict windows for TwoRoom, and context–goal pairs for T2. Evals. denotes predictive evaluations per plan. Repeats reuse the same panel; TEST denotes the official split. Checkpoints and randomness are specified in Appendix A.2.

Domain	Study	Panel	N	Evals.	H	h	Learned configuration / replication
PushT	Proposer T1	TEST	500	9,000	25	25	3 proposer training seeds
PushT	Proposer T2	TEST pairs	1,000	9,000	25	25	Validation-selected median run
PushT	Proposer T3	TEST	500	9,000	25	25	Validation-selected median run
TwoRoom	Proposer T1	Strict TEST	500	9,000	25	25	Proposer training seed 0
TwoRoom	Proposer T2	Strict TEST pairs	1,000	9,000	25	25	Proposer training seed 0
TwoRoom	Proposer T3	Strict TEST	500	9,000	25	25	Proposer training seed 0
Cube	LeCompass T1	TEST	1,000	192	25	5	Fixed learned components
Cube	Direct / Fixed-25 T1	TEST	1,000	9,000	25	25	No downstream LeCompass component
Cube	LeCompass T3	TEST	1,000	192	25	5	Same fixed learned components

All downstream task-trained components exclude official TEST episodes from gradient updates. PushT and TwoRoom proposers exclude their TEST lists from training and validation. CompassFlow excludes TEST episodes from training and calibration; retrieval banks do likewise. Cube readouts use 528 non-TEST training episodes, disjoint from the 1,000-episode T1/T3 panel and the full 2,500-episode TEST pool. Released LeWM pretraining lacks episode-level provenance. PushT inherits fixed LeWM action-normalization statistics that include its TEST episodes; these statistics remain shared across downstream studies (Appendix B.2).

A.2 Checkpoints and evaluation randomness

PushT proposer T1 reports three training seeds; T2/T3 use the checkpoint with median validation MSE across them. All TwoRoom proposer studies use the same TEST-excluded seed-0 checkpoint.

Table 5: **Search and attribution protocols.** N counts episodes for PushT/Cube and strict evaluation windows for TwoRoom. Evals. denotes predictive evaluations per plan; H/h gives the prediction horizon and nominal execution prefix in raw actions. $0/192$ denotes unranked execution versus IID best-of-192 predictive selection. Repeated runs reuse the same panel. Checkpoints and source-sampling seeds are specified in Appendix A.2.

Domain	Study	Panel	N	Evals.	H/h	Model / sampling repeats
PushT	Action-space CEM	TEST	500	192	25/5	Single fixed run
PushT	Source-CEM target contrast	TEST	500	192	25/5	3 trajectory-field seeds
PushT	Ranking attribution	TEST	500	0/192	25/5	Same 3 trajectory-field seeds
TwoRoom	Action-space CEM	Strict TEST	500	192	25/25	Single fixed run
TwoRoom	Flow executors	Strict TEST	500	192	25/25	3 trajectory-field seeds
TwoRoom	Ranking attribution	Strict TEST	500	0/192	25/25	Same 3 trajectory-field seeds
Cube	Ranking attribution	TEST	1,000	0/192	25/5	Fixed model/field; 3 source seeds
Cube	Target attribution	TEST	1,000	192	25/5	Fixed model/field; 3 source seeds

PushT and TwoRoom flow-based results report mean and sample standard deviation across three trajectory-field training seeds; repeated evaluations of a panel are not independent episodes. Cube attribution fixes the model and field and varies only source-sampling seeds 0, 1, and 2.

All rows in Table 4 use evaluation seed 0. Official TEST PushT search and ranking use source-noise base seed 0. TwoRoom flow executors use source-sampling seed 0 across field checkpoints and reuse the initial source coordinates across replans within each window. Primary PushT/TwoRoom ranking pairs use source-sampling seed 0; additional unranked source-seed repeats are reported only for TwoRoom. Appendix F specifies these comparisons and distinguishes Cube's matched IID suite from its headline Source-CEM run.

A.3 Budgets, replanning, and stopping

One predictive evaluation scores one candidate with frozen LeWM, after source decoding for CompassFlow. Horizon H and execution prefix h are measured in raw actions; all reported LeCompass controllers use $H = b\kappa = 25$, with $b = 5$ and $\kappa = 5$. PushT target-interface runs use $h = 25$; official TEST search arms use $h = 5$.

In official TEST PushT search, buffers are checked every raw step. Flow replans and refreshes the proposer when its five-action buffer empties or an auto-reset flushes it after termination or truncation. Action-space CEM replans every five actions but refreshes its proposer target only every 25, clearing the buffer then.

Environment-action caps are separate. For a fixed five-episode PushT batch $\mathcal{B}$, define $M_{\mathcal{B}} = \max_{i\in\mathcal{B}}(g_i - s_i)$ from demonstration start/goal indices s_i, g_i. PushT target-interface T1 uses $M_{\mathcal{B}}+50$ actions, T2 uses 250, and T3 uses $M_{\mathcal{B}} + 50 + L$ for L replacement actions. Official TEST search arms share ordered episodes and action caps. TwoRoom proposer T1/T2/T3 use 150/250/165 actions; executor and ranking studies use 150. Cube T1/T3 use 225.

Batched PushT evaluations and TwoRoom proposer studies latch success and continue to the cap with auto-reset. The TwoRoom executor stops at target entry or the cap. Cube native reach does not terminate execution: LeCompass stops at the third opening action's post-action check or the cap; Direct and Fixed-25 run to the cap.

A.4 Baselines and matched attribution arms

Fixed-temporal oracle. For demonstration start and goal indices s, g, Fixed-k uses observations at $s + k, s + 2k, \ldots < g$ and appends the final goal. Segment budgets equal successive demonstration-index differences from s. Elapsed environment time alone selects the active waypoint, clamped to the final goal after the schedule ends; reaching a waypoint does not advance it. Fixed-25 therefore has the same target schedule as Direct when $g - s \leq 25$.

This test-time demonstration access is unavailable to Direct and LeCompass. On Cube's 1,000-episode panel, Fixed-25 uses 3,966 intermediate-or-final observations from those episodes' expert

frames, not the separate 1,500-episode reserve partition. Fixed-k is undefined for cross-episode T2 under this same-episode construction.

Predictive-ranking attribution. Paired arms use matched source-noise construction within each field checkpoint and source seed (Appendix A.2). At a matched planning call, unranked execution selects the first decoded candidate without scoring; ranked IID selection scores all 192 candidates and selects the lowest-cost one. Both retain the learned prior conditioned on frozen JEPA history. Decoded trajectories may differ after closed-loop histories diverge. Cube additionally retains current- and goal-token readouts and the execution program.

Cube scoring-target attribution. Both arms retain ranked IID sampling with 192 candidates, the execution program, and the same current- and final-goal-token readouts. Only the scoring reference changes: final-goal latent versus proposed intermediate latent. The complete three-arm chain and its separation from the headline run are reported in Appendix F.

A.5 Perturbations and duration sensitivity

Table 6 defines the realized-execution perturbation used for each domain's T3 results.

Table 6: **T3 perturbation protocols.** Perturbations act on realized execution while the controller continues to observe and update.

Domain	Perturbation
PushT	Beginning at $\lfloor 0.4\, t_g \rfloor$, replace 15 emitted actions with actions sampled uniformly between each environment action-space bound; the inner policy continues to observe and update.
TwoRoom	The same 15-action perturbation beginning at $\lfloor 0.4\, t_g \rfloor$, evaluated with the strict collision-and-slide criterion.
Cube	After the stable-carry trigger, apply the bounded forced-opening intervention defined in Appendix C.

For PushT/TwoRoom, $t_g = g - s$ is the demonstration goal offset from evaluation start. Replacement actions use a perturbation-policy random-number generator seeded with evaluation seed 0, independently of checkpoint randomness. Cube's trajectory-triggered intervention uses no random sampling (Appendix C); all 1,000 episodes remain in the denominator, including those that never meet the stable-carry trigger.

The 15-action perturbation protocol was fixed before the longer-duration sensitivity sweep. Table 7 varies duration on the same 500-episode TEST panel, retaining start fraction 0.4 and adding the replacement count to the nominal action cap.

Table 7: **PushT T3 perturbation-duration sensitivity.** Successes out of 500 TEST episodes, with rates in parentheses. The prespecified primary protocol replaces 15 actions; the remaining rows vary duration at the same start fraction of 0.4.

Replaced actions	Direct	Fixed-25	State-indexed
15	21 (4.2%)	101 (20.2%)	145 (29.0%)
30	20 (4.0%)	50 (10.0%)	88 (17.6%)
45	20 (4.0%)	30 (6.0%)	61 (12.2%)
90	14 (2.8%)	26 (5.2%)	23 (4.6%)

State-indexed execution retains a positive point difference over Fixed-25 through 45 replacement actions; the difference is -0.6 percentage points at 90 actions.

A.6 Success criteria and statistical reporting

PushT. Native joint object-and-tool success is latched over post-action frames: combined pusher-and-block position error is below 20 pixels and wrapped block-angle error below $\pi/9$.

TwoRoom. The strict protocol models the agent as a disk, resolves motion by swept-disk collision detection and obstacle sliding, and requires agent-center distance to the target below 16 pixels.

Cube. Native reach is $\mathrm{ever}(d_{\mathrm{target}} \leq 0.04\,\mathrm{m})$, including frame zero; contact, pick, and carry are not prerequisites. Semantic completion additionally requires finished release and target containment in the post-action observation after the third opening command (Appendix C).

Fixed-seed Cube outcome intervals are episode-level Wilson intervals. Paired comparisons across domains use exact McNemar tests when reported.

B Target Proposal and Trajectory Search Implementation

B.1 Proposer training construction

The frozen encoder constructs latent sequences at stride five, matching the action-block size. Each sample contains three normalized history latents, a normalized later-goal latent, and span κ. The target shifts all three history positions by $H = 5\kappa$ raw actions; Eq. 5 averages their squared errors. The goal is sampled at or beyond the latest shifted position, with terminal-goal probability 0.5; context and goal remain in one trajectory. Table 8 gives the supported spans, architecture, and optimization.

Table 8: **Proposer training configuration.** The architecture and optimization settings are shared across PushT, TwoRoom, and Cube.

Component	Configuration
Input	3 history latents, later-goal latent, and span index
Output	3 shifted latent predictions
Sequence model	6-layer transformer, 16 heads, 64-D head
Latent dimension / MLP width	192 / 2,048
Span embedding	32-D; $\kappa \in \{2, \ldots, 7\}$
Dropout	0.1
Optimizer	AdamW, LR 3×10^{-4}, weight decay 10^{-3}
Schedule	2 warmup epochs, then cosine decay
Batch / training duration	1,024 / 30 epochs
Validation	10% episode split; checkpointing by validation MSE

Only the latest output is inverse-normalized with training-split statistics for frozen-LeWM scoring. All reported proposer-based controllers use $\kappa = 5$; target refresh follows the protocol-specific schedule in Appendix A.3, which can differ from replanning cadence.

B.2 Trajectory-prior training and action mapping

Normalized action chunks occupy a padded 35-action tensor; deployment uses the 25-action horizon $\kappa = 5$. With the interpolation from Eq. 6, let $v^\star = \bar{\mathbf{a}} - \epsilon$, and expand the validity mask M_κ across action channels, with $\bar{M}_\kappa = 1 - M_\kappa$. The implemented objective includes a padded-coordinate penalty:

$$\mathcal{L}_{\mathrm{flow}} = \frac{\|M_\kappa \odot (v_\phi - v^\star)\|_2^2}{\|M_\kappa\|_1} + 0.01\,\frac{\|\bar{M}_\kappa \odot v_\phi\|_2^2}{\max(\|\bar{M}_\kappa\|_1, 1)}.$$

Source noise and interpolation time are sampled independently per chunk. Padded source coordinates are zero, and the mask is re-applied during transport. Table 9 gives training and sampling settings.

After eight Euler steps, the valid normalized chunk is inverse-scaled and processed by the domain's raw-candidate rule (Table 10). The resulting raw trajectory is retained for execution; a copy standardized in LeWM's convention is scored. The selected raw prefix then enters the execution interface.

TwoRoom and Cube fit field action scalers on non-TEST training splits and store them with field artifacts. PushT shares the inherited LeWM action statistics between field training and scoring. At evaluation, the LeWM standardizer is reconstructed from the fixed raw-data artifact recorded in run provenance, not refitted to the evaluation panel.

Table 9: **Trajectory-field training configuration.** Action dimension is 2 for PushT and TwoRoom and 5 for Cube.

Component	Configuration
Conditioning	3 normalized 192-D history latents; 32-D span embedding; no target input
Velocity network	512-D condition MLP + 128-D sinusoidal-time MLP + 2-layer 512-D action MLP
Capacity / deployed horizon	35 / 25 raw actions; supported $\kappa \in \{2, \ldots, 7\}$; deployed $\kappa = 5$
Optimization	Adam, LR 3×10^{-4}, weight decay 0; batch 256; 8,000 steps
Objective	Masked rectified-flow MSE on valid positions + 0.01 auxiliary padded-coordinate penalty
Sampling	8 Euler steps
Training repeats	PushT and TwoRoom executor studies: field seeds 0, 1, and 2; Cube: fixed seed-0 field
Checkpoint	Final training step; no early stopping

Table 10: **Pre-scoring action bridge.** Invert the trajectory-field scaler, apply the raw-candidate rule, then standardize a copy for frozen-LeWM scoring. The raw candidate is retained as the planned trajectory and passed to the execution interface after selection.

Domain	Dim.	Raw candidate rule	LeWM action mapping
PushT	2	No candidate clipping	$(a - \mu_{\mathrm{wm}})/s_{\mathrm{wm}}$
TwoRoom	2	Clip each channel to $[-1, 1]$	$(a - \mu_{\mathrm{wm}})/s_{\mathrm{wm}}$
Cube	5	Clip each channel to $[-1, 1]$	$(a - \mu_{\mathrm{wm}})/s_{\mathrm{wm}}$

B.3 Source-space predictive trajectory search

PushT and TwoRoom executor studies compare ranked IID sampling with Source-CEM. Ranking attribution in all domains and Cube target attribution use ranked IID; headline Cube T1/T3 uses Source-CEM.

Initial banks draw IID standard-normal coordinates on valid positions, with padded positions fixed at zero. PushT and Cube allocate banks per planning call; TwoRoom allocates per window and reuses source coordinates across replans. Reused coordinates are decoded under the new observed history, not reused as action trajectories.

Ranked IID sampling. Decode 192 source tensors under the current history, score each once with frozen LeWM, and send the lowest-cost raw prefix to the execution interface. Unranked execution selects the first decoded candidate without scoring. Neither arm adapts its source distribution.

Source-CEM. The same 192-evaluation budget is divided into three 64-candidate rounds:

1. Decode and score the initial 64 sources; fit a diagonal Gaussian to the eight lowest-cost sources in that round.
2. Clamp the fitted mean to $[-1, 1]$ and standard deviations to $[0.3, 1]$.
3. In each of two adaptive rounds, evaluate eight standard-normal prior anchors and 56 adaptive candidates (the mean and 55 samples), clamping evaluated source coordinates to $[-3, 3]$. Refit and clamp using that round's eight elites.
4. Return the lowest-cost decoded trajectory across all three rounds.

The fitted distribution is not warm-started across calls. Prior anchors retain base-distribution coverage; adaptive proposals concentrate evaluations around low-cost sources.

C Cube Execution Program

Cube's readouts, state machines, channel ownership, and recovery are specified below. Evidence predicates, temporal thresholds, and recovery limits were determined from development rollouts and fixed before evaluation.

C.1 READOUTS AND THEIR EXECUTION ROLES

Acquisition predicts world-frame block-minus-end-effector XY displacement from current patch tokens. Placement uses current and cached final-goal tokens to predict target-minus-block XY displacement $\widehat{\Delta xy}$ and block height. The controller derives lateral commitment and set-down decisions from these estimates, using height for vertical correction and descent/set-down eligibility. Execution memory gates release from sustained contact and lateral alignment, independently of the placement controller's set-down phase.

Final-goal tokens are encoded once per episode and are independent of the intermediate planning target. One placement estimate is shared by memory and the placement controller. Readouts are adapted sequentially before evaluation and then fixed (Appendix D).

C.2 STATE MACHINES AND NOMINAL PROGRESSION

Table 11 lists two coupled state machines. The ten-state execution memory (hereafter phase memory) tracks contact persistence, grip/release eligibility, and recovery. It reads the observed scalar proprioceptive gripper-contact signal and $\widehat{\Delta xy}$, not simulator block/target coordinates or evaluator success; held contact means a signal above 0.5. The eight-state placement controller implements spatial ownership and set-down geometry, not additional task-level subgoals.

Nominal execution follows ACQUIRE→CARRY→TARGETHOLD→RELEASE. ACQUIRE groups SEEK, ACQUIREPENDING, and ACQUIREASSIST; RELEASE groups the opening sequence and terminal RELEASED state. Other memory states handle loss, recovery, and exhaustion.

Table 11: **Coupled Cube state machines.** Execution memory maintains interaction evidence and recovery state; the placement controller determines spatial action ownership. Narrative task phases group execution-memory states.

Machine	States	Primary responsibility
Execution memory	SEEK, ACQUIREPENDING, ACQUIREASSIST, CARRY, LOST, SEEKRECOVERY, TARGETHOLD, RELEASE, RELEASED, RECOVERYEXHAUSTED	Contact persistence, grip/release gating, bounded recovery
Placement controller	PREPICK, TRANSPORT, CENTER, DESCEND, COMMITSETDOWN, AWAITHANDOVER, RELEASE, STABILIZE	Spatial channel ownership and set-down realization

Confirmation and release. Within one consecutive held-contact run, three observations activate bootstrap grip protection and twelve confirm CARRY; earlier loss resets acquisition. Assisted acquisition expires after 60 steps and returns to ACQUIREPENDING. In CARRY, held contact and $\|\widehat{\Delta xy}\|_2 \leq 0.012\,\mathrm{m}$ for two consecutive observations confirm TARGETHOLD. Six further observations satisfying the same predicate initiate release; failure resets this dwell counter.

RELEASED denotes a completed release-command sequence, not semantic success. T1 disables recovery, allows one carry activation, and stops after the third opening action or at timeout. Semantic completion is checked in that action's post-action observation, with no extra settling window (Appendix A.6).

C.3 ACTION OWNERSHIP AND ACTUATOR ORDER

LeWM scores five-channel planner candidates before execution-side assignments. After trajectory selection, the placement controller assigns action channels according to Table 12. In PREPICK, acquisition supplies x, y. The controller/actuator-composed action is not re-scored.

After channel assignment, the grip-floor actuator enforces a fixed lower bound on the gripper-closing command in ACQUIREASSIST, CARRY, and TARGETHOLD to help maintain the grasp; otherwise the planner's pre-pick gripper value can pass through. The release actuator then overrides only the gripper with -1.0 for three actions. Release and grip-floor states are disjoint. The nominal order is

Table 12: **Cube action-channel ownership before actuator overrides.** In PREPICK, acquisition supplies x, y; the planner supplies z, yaw, and the base gripper action. In the other seven phases, placement supplies all but yaw. Grip-floor and release actuators act afterward.

Placement phase	x	y	z	yaw	Gripper base
PREPICK	Acquisition	Acquisition	Planner	Planner	Planner conditional grip floor
Other seven phases	Placement	Placement	Placement	Planner	Placement

planner, placement controller, grip floor, release actuator, then environment. Appendix G reports actual T1 value passthrough separately from assigned ownership.

C.4 BOUNDED RECOVERY

Recovery-enabled T3 retains the learned components and nominal parameters, permits one recovery attempt, and allows at most two carry activations. Only CARRY admits recovery: three consecutive unheld-contact observations lead through LOST to SEEKRECOVERY if budget remains. Rollback atomically discards the unexecuted prefix and resets the placement controller to PREPICK and the acquisition servo, preserving environment time, observed latent history, and outer-loop task progress. A new contact segment with twelve confirming observations activates the second CARRY segment and resumes replanning. Loss after TARGETHOLD does not enter recovery; exhaustion enters absorbing RECOVERYEXHAUSTED.

C.5 T3 FORCED-OPENING INTERVENTION

After five consecutive CARRY steps with held contact, the perturbation controller overrides only the gripper to -1.0, after nominal controllers and actuators and before environment execution. Each new observation reports contact from the preceding action. Before issuing another forced action, the override stops if three consecutive unheld observations confirm loss, the phase leaves CARRY, or three forced-opening actions have been emitted. Reaching the cap without confirmed loss is a failed induction; later actions resume the nominal chain. The intervention uses no random sampling.

D CLOSED-LOOP READOUT ADAPTATION AND DIAGNOSTICS

This section supports the spatial-feedback analysis in Sec. 7.3; the execution rules are specified in Appendix C.

D.1 ADAPTATION AND CHECKPOINT SELECTION

Readout adaptation uses 528 training, 113 validation, and 113 heldout episodes. Expert supervision and rollout aggregation use training episodes; validation states support diagnostics and selection, not gradient updates. The heldout partition is unused here. The JEPA encoder and predictor remain frozen.

Acquisition starts with expert frames, then combines expert data with rollout states sampled uniformly across no-correction, pixel-based reference, and initial token-based controllers. The pixel reference estimates geometry directly from images. Further acquisition adaptation emphasizes near-field and false-zero hard negatives while retaining successful and far-field frames. Acquisition is then frozen while placement is adapted on expert frames and training rollouts from the initial token-based, pixel-based, and oracle controllers. Separate validation rollouts from these three sources support placement diagnostics and selection. Table 13 specifies the deployed heads.

Table 13: **Deployed Cube readouts.** Simulator state supplies training labels; deployed readouts consume frozen visual tokens. Both heads are fixed before TEST evaluation.

	Acquisition (R2)	Placement (P1)
Input and target	Current $16\times16\times192$ patch tokens; world-frame $xy_{\text{block}} - xy_{\text{effector}}$ (2 values).	Current and cached final-goal token grids; world-frame $xy_{\text{target}} - xy_{\text{block}}$ and block height above rest (3 values).
Network	1×1 convolutions $192 \to 128 \to 32$, spatial softmax to 32 keypoint coordinates (64 values), then a 256–256 MLP.	The same spatial trunk for each input, with separate weights; concatenated features feed a 256–256 MLP. No learned phase-gate head.
Loss and optimization	Huber loss ($\delta = 1$) on targets divided by $0.05\,\text{m}$; AdamW, learning rate 10^{-3}, weight decay 10^{-4}, OneCycle schedule; 8,000 steps, batch 128.	Same scaled Huber loss and AdamW settings, 8,000 steps, batch 128; constant learning rate.
Training-state sampling	Choose among four equally weighted sources (expert, no-correction, pixel-controller, token-controller), then sample an episode and a frame; within-episode frame weights favor near-field and false-zero cases.	Choose among four equally weighted sources (expert, initial-token, pixel-controller, oracle), then sample an episode, phase, and frame.

The deployed acquisition and placement checkpoints were selected using offline validation diagnostics. Selection was finalized before the corresponding closed-loop outcomes were observed, and the selected readouts remained fixed throughout formal evaluation.

D.2 Readout errors across evaluation distributions

Expert-frame diagnostics use the 113-episode validation partition: 3,830 acquisition and 9,469 placement frames. Controller-induced diagnostics use 5,338 frames from 40 validation episodes under the expert-frame-trained acquisition readout, and 5,580 failure-state frames from 31 validation episodes under the initial placement readout. These stored states support offline diagnostics and selection, not training. Table 14 compares readouts on identical frames within each rollout-state pair.

Table 14: **Readout errors on expert and controller-induced states.** Expert and rollout rows use distinct distributions; readouts within each rollout pair use identical frames. p50/p90 summarize per-frame world-frame XY Euclidean errors (mm): block minus effector for acquisition, target minus block for placement.

Stage	Readout	Evaluation states	Episodes	p50	p90
Acquisition	Expert-frame trained	Expert validation	113	2.62	5.22
Acquisition	Rollout adapted	Expert validation	113	3.81	10.12
Acquisition	Expert-frame trained	Self-induced rollout	40	62.98	133.09
Acquisition	Rollout adapted	Same 5,338 rollout frames	40	9.23	42.31
Placement	Initial	Expert validation	113	5.71	11.17
Placement	Rollout adapted	Expert validation	113	5.90	11.22
Placement	Initial	Self-induced failure states	31	190.46	733.50
Placement	Rollout adapted	Same 5,580 rollout frames	31	11.86	21.30

Across no-correction, pixel-reference, and initial-token rollout states, acquisition median error changes from 49.44, 14.09, and 62.98 mm to 10.10, 7.34, and 9.23 mm, respectively.

On the same 5,338 acquisition rollout frames, wrong-direction predictions fall from 2,420 (45.3%) to 118 (2.2%). Wrong direction means a negative dot product with the true correction. Predicted or true vectors with norm at most $10^{-9}\,\text{m}$ are excluded; all 5,338 frames remain eligible for both readouts.

D.3 CLOSED-LOOP VALIDATION

On the same 113-episode validation panel, acquisition adaptation increases confirmed picks from 47/113 (41.6%; 95% Wilson interval $[32.9, 50.8]$) to 93/113 (82.3%; $[74.2, 88.2]$), with 48 improved and two worsened pairs (exact McNemar $p = 2.27 \times 10^{-12}$).

Holding adapted acquisition fixed yields the same 93 confirmed-pick episodes for both placement configurations. Within this cohort, native reach increases from 11/93 (11.8%; $[6.7, 19.9]$) to 92/93 (98.9%; $[94.2, 99.8]$), with 81 improved and zero worsened pairs (exact McNemar $p = 8.27 \times 10^{-25}$). Off-target set-downs decrease from 36/93 (38.7%; $[29.4, 48.9]$) to 3/93 (3.2%; $[1.1, 9.1]$). Off-target set-down is one of six mutually exclusive placement-outcome categories; the six categories sum to 93 episodes in each configuration. Native reach and off-target set-down are not mutually exclusive.

D.4 REPRESENTATION AND INPUT INTERVENTIONS

Planning-latent probes. Ridge and MLP probes use frozen 192-dimensional projected CLS latents, with 71,232 training and 14,656 validation frames. Both decode end-effector XY more accurately than block-relative XY (Table 15 and Fig. 5); height is not tested.

Table 15: **Spatial probes of the frozen planning latent.** All scores use the same 14,656 validation frames. Median errors are world-frame XY Euclidean distances.

Prediction target	Ridge R^2	MLP R^2	MLP median error (mm)
End-effector XY	0.946	0.957	16.3
Block XY	0.334	0.294	89.7
Block minus end-effector XY	−0.036	−0.128	90.7

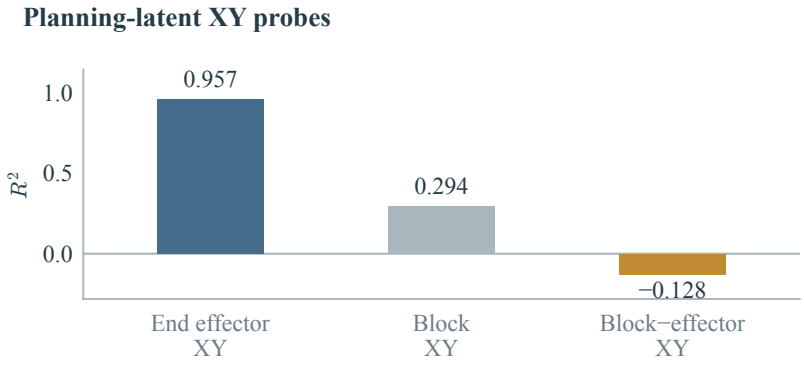

Figure 5: **Projected-CLS planning-latent XY probes.** MLP R^2 on the same 14,656 validation frames as Table 15. The probes assess XY readout, not height or action-conditioned ranking; poor decoding does not establish that the latent lacks the information.

For relative XY, a zero-displacement predictor has median error 73.1 mm, versus 5.68 mm for a development-only pixel CNN on the same frames. These are median-distance comparisons; R^2 instead compares mean-squared error with a mean-label predictor.

Patch-token localization. Each intervention swaps one 16×16 pixel cell between two frames, re-encodes both, and ranks 256 tokens by change magnitude. Five of twelve swaps have identical pixels and are excluded. The corresponding token ranks first in all seven non-null swaps.

Readout input interventions. Initial, pre-adaptation readouts are tested with changed inputs but fixed labels and weights. Independently shuffling token positions within each sample raises acquisition expert-frame p90 error from 5.22 to 253.32 mm. For placement, swapping current/goal

inputs, replacing the goal with the current input, and shuffling tokens raise p90 error from 11.17 mm to 149.33, 126.19, and 188.02 mm, respectively.

E Frozen-Latent Target-Geometry Diagnostics

E.1 Distance distributions and summary scales

The diagnostic in Sec. 7.1 uses three unsquared L_2 distance distributions in each frozen checkpoint. $\mathcal{A}$ measures within-episode anchor-to-successor distances over one latent step (five raw frames). $\mathcal{G}$ uses uniformly sampled pairs of distinct retained anchor-frame entries, including within- and cross-episode pairs. $\mathcal{E}$ measures error at the proposer's latest inverse-normalized output:

$$d_t^{\text{prop}} = \left\| P_\psi(\mathbf{z}_t^{\text{hist}}, z_{\text{final}}, \kappa) - z_{t+\kappa} \right\|_2, \qquad \kappa = 5. \tag{7}$$

Here t and κ index latent steps. History contains the anchor and two preceding latents; the supervised future is 25 raw frames ahead, whereas $\mathcal{A}$ spans five.

Define $A = \text{median}(\mathcal{A})$, $G = q_{0.95}(\mathcal{G})$, and $E = \text{median}(\mathcal{E})$. Normalizing all distances by G preserves within-checkpoint ordering and gives the summary ratios

$$R_{\text{usable}} = 1 - \frac{A}{G}, \qquad R_{\text{error}} = \frac{E/G}{1 - A/G}. \tag{8}$$

Thus $R_{\text{error}} = E/(G - A)$ is the main-text distance ratio, not a dynamical-reachability test.

E.2 Anchor sampling and uncertainty

For each of 754 validation episodes per environment, we request eight evenly spaced anchors with the required history and five-step future. Shorter valid ranges contribute every available position without duplication. Each anchor contributes one $\mathcal{A}$ and one $\mathcal{E}$ sample; equally many global pairs form $\mathcal{G}$. Each distribution has 5,994 samples for PushT, 6,032 for Cube, and 5,845 for TwoRoom.

Uncertainty uses 2,000 episode-level bootstrap resamples. Each resample retains the selected episodes' anchors, redraws global pairs from the resampled anchor pool, and recomputes both ratios from its medians and global-pair 95th percentile. The reported 95% intervals are the 2.5th and 97.5th percentiles of these ratio draws.

E.3 Distributions and cross-domain comparison

Figures 6 and 7 show raw and checkpoint-normalized empirical CDFs in the unnormalized 192-dimensional projected CLS space. Table 16 reports the summary ratios. PushT and Cube have median local-transition distances and proposer errors well below the global-pair scale; TwoRoom has narrower local-to-global separation relative to proposer error.

Table 16: **Relative target-geometry scales.** Ratios use unrounded statistics; brackets are 95% episode-bootstrap intervals. $R_{\text{error}} = E/(G - A)$ is a distance ratio and may exceed one.

Environment	A/G	E/G	R_{usable} [95% CI]	R_{error} [95% CI]
PushT	0.243	0.104	0.7573 [0.7515, 0.7620]	0.1370 [0.1327, 0.1422]
Cube	0.255	0.169	0.7446 [0.7404, 0.7470]	0.2274 [0.2231, 0.2329]
TwoRoom	0.791	0.610	0.2087 [0.2026, 0.2152]	2.9238 [2.8229, 3.0240]

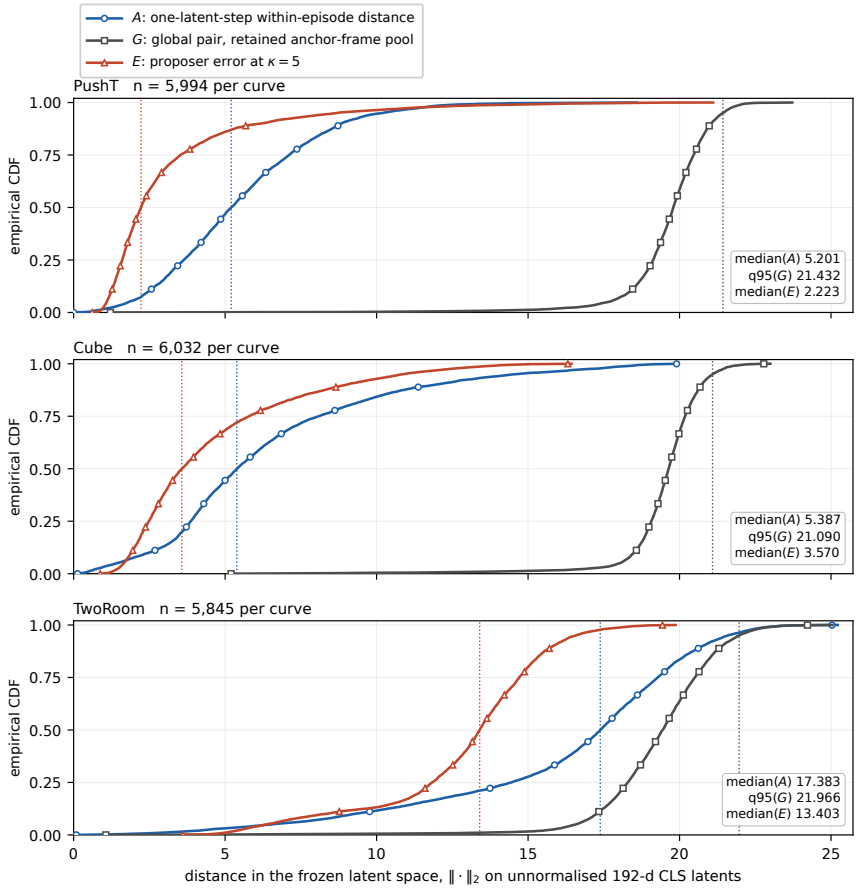

Figure 6: **Frozen-latent distance distributions.** Empirical CDFs of one-latent-step distance $\mathcal{A}$, global-pair distance $\mathcal{G}$, and proposer error $\mathcal{E}$ (labeled A, G, and E) for independently trained domain checkpoints. Distances are unsquared L_2 in unnormalized 192-D projected CLS space. Dotted lines mark $A = \mathrm{median}(\mathcal{A})$, $G = q_{0.95}(\mathcal{G})$, and $E = \mathrm{median}(\mathcal{E})$.

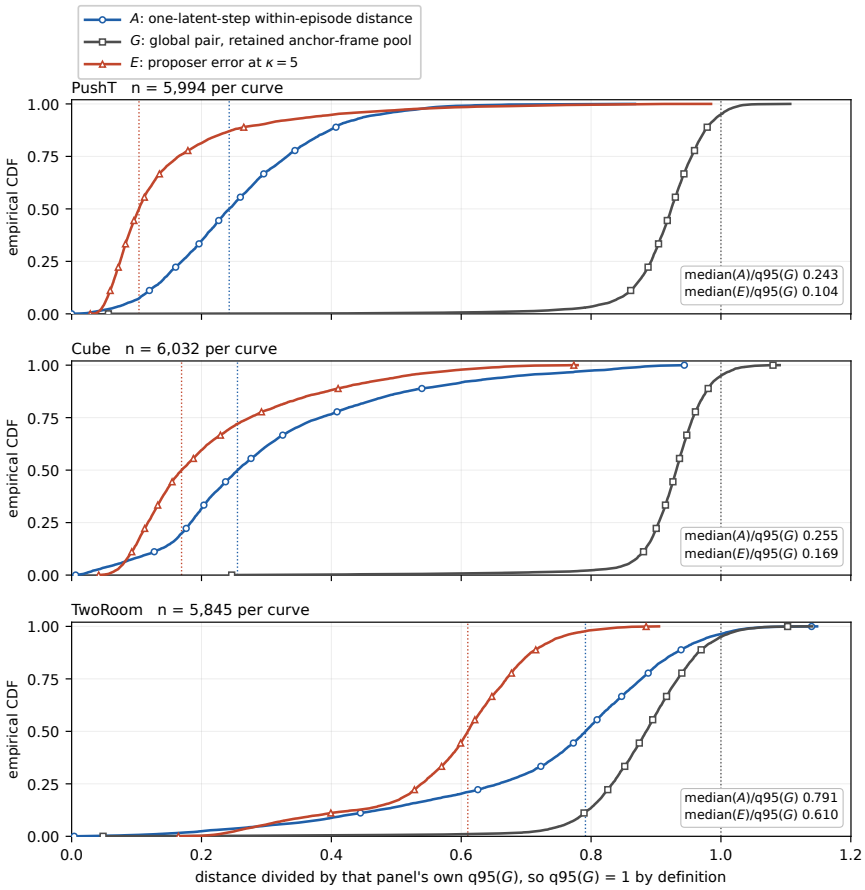

Figure 7: **Checkpoint-normalized distances.** All $\mathcal{A}$, $\mathcal{G}$, and $\mathcal{E}$ distances are divided by the checkpoint-specific $q_{0.95}(\mathcal{G})$. This sets the 95th-percentile global-pair reference to one for cross-checkpoint scale comparison without changing within-checkpoint distance ordering.

F Executor and Attribution Breakdowns

The comparisons below separate search configuration, scoring target, and predictive ranking (Secs. 6.2 and 7.2). Protocols and source-bank construction are specified in Appendices A and B.3. Repeated checkpoints or source seeds are not additional independent episodes.

F.1 Official TEST PushT search and refinement

Table 17 uses the same ordered 500 TEST episodes, frozen LeWM and proposer, 192 evaluations per plan, and nominal five-action prefix. The single action-space CEM run refreshes its target every 25 actions and warm-starts across calls. Source-CEM uses three field checkpoints, refreshes every five actions, and refits each call.

Table 17: **Official TEST PushT configuration and target contrasts.** Each checkpoint reuses the same 500 episodes. Differences are $B - A$ (Source-CEM proposed minus action CEM) and $B - C$ (proposed minus final-goal Source-CEM), in percentage points; 95% intervals use 10,000 episode-paired bootstrap resamples. The repeated A column is one fixed baseline run.

Field seed	Action CEM (A)	Source proposed (B)	Source final (C)	$B - A$ [95% CI]	$B - C$ [95% CI]
0	122	264	76	$+28.4\ [22.8, 33.8]$	$+37.6\ [32.6, 42.6]$
1	122	274	64	$+30.4\ [24.8, 35.8]$	$+42.0\ [36.8, 47.2]$
2	122	249	68	$+25.4\ [20.0, 30.6]$	$+36.2\ [31.0, 41.4]$

Run-level predictive-evaluation totals are 5.67–5.79 million for proposed-target Source-CEM and 4.84–4.93 million for final-goal Source-CEM; batch logs bound action-space CEM at 4.04–9.39 million.

Ranked IID and proposed-target Source-CEM share the panel, checkpoint, model, proposer, scoring target, prefix, and per-plan budget; only source sampling is refined (Table 18). Across the three field checkpoints, success is $44.9 \pm 1.1\%$ for ranked IID and $52.5 \pm 2.5\%$ for Source-CEM (mean $\pm$ sample standard deviation).

Table 18: **Official TEST PushT source refinement.** Each row pairs 500 episodes within a field checkpoint. Differences are Source-CEM minus ranked IID (percentage points); 95% intervals use 10,000 episode-paired bootstrap resamples. p is two-sided exact McNemar. Checkpoints are not pooled as independent episodes.

Field seed	Source-CEM	Ranked IID	Difference [95% CI]	Source-only / IID-only	Exact p
0	264/500 (52.8%)	228/500 (45.6%)	$+7.2$ $[2.0, 12.2]$	106 / 70	.00815
1	274/500 (54.8%)	227/500 (45.4%)	$+9.4$ $[4.2, 14.8]$	119 / 72	.000828
2	249/500 (49.8%)	218/500 (43.6%)	$+6.2$ $[1.0, 11.4]$	107 / 76	.0263

F.2 Official TEST TwoRoom executor and refinement

On 500 strict TEST TwoRoom windows with the proposed target and a nominal 25-action prefix, ranked IID reaches $77.3 \pm 0.4\%$ and Source-CEM $77.3 \pm 1.0\%$ across three field checkpoints. Table 19 gives per-checkpoint counts, including final-goal Source-CEM. Matched Source-CEM-minus-ranked-IID effects are -0.8, 0.0, and $+0.8$ percentage points across field checkpoints 0–2 (Table 20). A two-level sensitivity bootstrap with 200,000 resamples over field checkpoints and paired windows gives a 95% interval of $[-2.3, 2.3]$ points for the mean difference. Equal aggregate successes do not imply identical successful windows.

Table 19: **Official TEST TwoRoom executor outcomes.** Counts use the same 500 strict windows for field seeds 0–2. Aggregates sum repeated evaluations, not independent windows. Source-sampling seed is 0.

Domain (N per seed)	Configuration	Seed 0	Seed 1	Seed 2	Aggregate
TwoRoom (500)	Ranked flow sampling + proposed latent	385	389	386	1,160/1,500
TwoRoom (500)	Source-CEM + proposed latent	381	389	390	1,160/1,500
TwoRoom (500)	Source-CEM + final-goal latent	310	300	320	930/1,500

Table 20: **Official TEST TwoRoom source refinement.** Source-CEM minus ranked IID on the same 500 strict TEST windows within each field checkpoint. Improved/worsened windows succeed only under Source-CEM/ranked IID. Intervals use 100,000 paired percentile-bootstrap resamples; p is two-sided exact McNemar.

Field seed	Difference (pp)	Improved/worsened	95% CI (pp)	p
0	-0.8	44/48	$[-4.6, 3.0]$	.7547
1	0.0	45/45	$[-3.8, 3.8]$	1.0000
2	$+0.8$	44/40	$[-2.8, 4.4]$	.7436

F.3 Predictive-ranking attribution on PushT and TwoRoom

PushT. Table 21 compares proposed-target IID best-of-192 selection with matched unranked execution on the same 500 TEST episodes, within each field checkpoint. Both retain the encoder and prior and share source coordinates at matched planning points; decoded trajectories can diverge with observed histories. Only the ranked arm scores candidates.

TwoRoom. Table 22 pairs proposed-target IID ranking and unranked execution at cadence 25 and source seed 0 on 500 strict TEST windows. These attribution runs are separate from the executor runs in Table 19.

Table 21: **Official TEST PushT predictive ranking.** Rows pair 500 episodes within each field checkpoint at source seed 0. Differences are ranked minus unranked (percentage points). Intervals use 10,000 episode-paired bootstrap resamples; p is two-sided exact McNemar. Repeats are not independent episodes.

Field seed	Best of 192	Unranked	Difference [95% CI]	Ranked-only / unranked-only	Exact p
0	228/500 (45.6%)	27/500 (5.4%)	$+40.2$ $[35.6, 44.8]$	212 / 11	2.07×10^{-49}
1	227/500 (45.4%)	27/500 (5.4%)	$+40.0$ $[35.2, 44.6]$	211 / 11	3.93×10^{-49}
2	218/500 (43.6%)	16/500 (3.2%)	$+40.4$ $[36.0, 44.8]$	206 / 4	9.76×10^{-56}

Table 22: **TwoRoom predictive ranking at cadence 25.** Matched source construction and seed 0 on 500 strict TEST windows; ranked IID selection uses the proposed target. Bootstrap 95% intervals and exact McNemar tests pair windows within each field checkpoint.

Field seed	Best of 192	No predictive ranking	Paired difference	Paired bootstrap 95% CI	McNemar p
0	387/500 (77.4%)	152/500 (30.4%)	+47.0 pp	$[42.6, 51.4]$	6.49×10^{-68}
1	390/500 (78.0%)	178/500 (35.6%)	+42.4 pp	$[37.8, 47.0]$	1.30×10^{-57}
2	388/500 (77.6%)	177/500 (35.4%)	+42.2 pp	$[37.6, 46.8]$	1.49×10^{-54}
Mean±SD	77.7±0.3%	33.8±2.9%	+43.9±2.7 pp	–	–

Unranked source seeds 0–2 yield success ranges of 11.4–30.4%, 20.8–35.6%, and 16.4–35.4% for field checkpoints 0–2. The prespecified matched comparison uses source seed 0.

F.4 CUBE PREDICTIVE-RANKING AND TARGET ATTRIBUTION

The Cube IID suite has three arms: first-candidate unranked execution (A), final-goal best-of-192 ranking (B), and proposed-target best-of-192 ranking (C). All share the 1,000-episode panel, one fixed model and field, and the execution interface. Thus $B - A$ adds final-goal ranking, $C - B$ changes its scoring target, and $C - A$ measures the net effect relative to unranked execution.

Source seeds 0–2 vary sampling only; seed 0 is primary and seeds 1–2 test directional sensitivity. The headline T1 run instead uses Source-CEM with source-seed base 7.

Table 23: **Cube IID ranking and scoring-target attribution.** Native 4-cm reach on the same 1,000 episodes. Intervals and exact McNemar tests are episode-paired within source seed. Mean±SD summarizes these three source-seed runs.

Absolute native-reach rates

Source-sampling seed	A: Unranked	B: Final goal	C: Proposed target
0	853/1000 (85.3%)	846/1000 (84.6%)	872/1000 (87.2%)
1	865/1000 (86.5%)	837/1000 (83.7%)	872/1000 (87.2%)
2	873/1000 (87.3%)	852/1000 (85.2%)	874/1000 (87.4%)
Mean±SD	86.4±1.0%	84.5±0.8%	87.3±0.1%

Episode-paired effects: difference [95% CI], exact McNemar p

Source-sampling seed	B−A: final-goal ranking	C−B: target replacement	C−A: net vs. unranked
0	-0.7 $[-2.8, 1.3]$, $p = .566$	$+2.6$ $[0.7, 4.5]$, $p = .0095$	$+1.9$ $[0.0, 3.8]$, $p = .059$
1	-2.8 $[-4.8, -0.9]$, $p = .0078$	$+3.5$ $[1.4, 5.6]$, $p = .0016$	$+0.7$ $[-1.1, 2.5]$, $p = .52$
2	-2.1 $[-3.8, -0.4]$, $p = .020$	$+2.2$ $[0.2, 4.1]$, $p = .033$	$+0.1$ $[-1.9, 2.0]$, $p = 1.0$
Mean±SD	-1.9 ± 1.1 pp	$+2.8 \pm 0.7$ pp	$+0.9 \pm 0.9$ pp

Final-goal ranking has negative native-reach point differences, while target replacement has positive differences, in all three runs (Table 23). Semantic-completion differences follow the same directions: $B - A$ is $-1.9, -3.0, -1.8$ points; $C - B$ is $+3.0, +3.1, +2.0$; and $C - A$ is $+1.1, +0.1, +0.2$. Small net effects coexist with high success under the shared prior and goal-aware execution interface.

G CUBE OUTCOME AND RECOVERY BREAKDOWNS

G.1 T1 OUTCOMES AND TRAJECTORY DIAGNOSTICS

Both fixed 500-episode partitions use the same frozen full-episode T1 configuration. Table 24 gives partition outcomes and their count-based micro-average; native reach is 86.9% with a 95% episode-level Wilson interval of $[84.7, 88.9]\%$.

Table 24: **Cube T1 partition outcomes.** Native reach and semantic completion are reported separately and combined by count across the two fixed 500-episode partitions.

Partition	Native reach	Semantic completion
Official main	440/500 (88.0%)	433/500 (86.6%)
Legacy windowed-protocol partition	429/500 (85.8%)	427/500 (85.4%)
Micro-average	869/1,000 (86.9%)	860/1,000 (86.0%)

Table 25 separates contact, geometric pick, lift, and reach. Lift is relative to the demonstration's fifth-percentile block height; maximum lift uses the retrospective descriptive threshold > 2 cm. Rollout-earned reach requires starting outside the target and first entering after execution begins, with all 1,000 episodes as denominator.

Table 25: **Cube T1 trajectory diagnostics on the common 1,000-episode panel.** Native reach includes frame zero; rollout-earned reach first occurs after execution begins. Geometric true pick requires contact > 0.5 and block lift ≥ 5 cm in the same observation.

Method	Any contact	Geometric true pick	Maximum lift > 2 cm	Native reach	Rollout-earned reach
Direct	569	0	0	37	1
Fixed-25	643	1	1	43	7
LeCompass	959	820	893	869	833

Baseline failures occur primarily between contact and geometric pick. Paired native-reach tests give exact McNemar $p < 10^{-200}$ for LeCompass versus either baseline; Fixed-25 versus Direct is unresolved ($p = .0703$). Rollout-earned reach is 83.3%, 0.1%, and 0.7% for LeCompass, Direct, and Fixed-25, respectively, excluding frame-zero entries as an explanation of the gap. The baseline-to-LeCompass nominal B/h ratio is $9.4\times$ (Table 3(a)).

The 1,000 T1 episodes comprise 860 semantic completions, 80 with contact but no geometric pick, 41 with no contact, 17 with post-pick drop or absent carry, and two with native reach but incomplete release. Completion is assigned first; other episodes follow the first failed check in the order contact, geometric pick, retained carry, native reach, release, and containment. These mutually exclusive classes cover the full panel, not native-reach successes. Geometric pick is not part of the completion criterion.

Table 26 summarizes block motion on the same panel.

Table 26: **Cube T1 block motion.** Medians over all 1,000 episodes, in meters. Tiny nonzero baseline values reflect the simulator numerical floor, not earned lift.

Method	Block XY path p50	Maximum block lift p50
Direct	5.7×10^{-14}	4.0×10^{-5}
Fixed-25	8.7×10^{-14}	4.0×10^{-5}
LeCompass	0.286195	0.103273

G.2 REALIZED ACTION PASSTHROUGH

Of 82,910 headline T1 actions, 57,520 (69.4%) occur in the placement controller's PREPICK state; other placement-controller states account for 30.6%. Planner z passes through on all pre-pick actions and yaw on all 82,910 actions. Pre-pick gripper commands pass through on 51,531/57,520 actions;

the grip floor changes 5,989. Planner x, y values almost never pass through because readouts supply them. These are observed value-passthrough counts, not causal channel contributions.

Among 820 geometric-pick episodes, median first-pick time is step 34 and the median fraction of observations at or after pick is 41.7%, including the pick observation.

G.3 HEIGHT-INFORMED SIMPLE-CONTROL REFERENCE

To test whether a simple controller can operate the Cube execution stack without learned action proposals or predictive scoring, we compare an unranked learned-prior arm with a height-informed rule controller on the same ordered 1,000-episode official TEST panel. Both use the same frozen patch-token readouts, observation-based memory, placement controller, release logic, goal frames, 225-action cap, and paper T1 success rule. The rule replaces the unranked arm's learned trajectory prior and additionally reads the non-privileged z coordinate of proprioceptive end-effector position; neither arm uses predictive candidate scoring or proposer-driven candidate selection. Consequently, this is a more-informed control reference, not a same-input ablation of the learned prior. The test run follows rule design and functional validation on a disjoint 113-episode development panel.

The unranked learned-prior arm is source-sampling seed 0 of the separate Cube attribution suite, not the Source-CEM headline configuration. Neither arm uses predictive candidate scoring, and their paired difference does not estimate a difference from the headline LeCompass run.

In PREPICK, the rule controller commands raw vertical action -0.3 with the gripper open (raw -1.0) until observed end-effector height is at most 0.026 m. It then commands closure (raw $+1.0$) and zero vertical action; after contact is observed, it commands raw vertical action $+0.6$ until the observed end-effector height is at least 0.06 m above its value at first contact. It maintains a closed-gripper command until downstream placement and release take ownership, and commands zero yaw throughout. The existing acquisition readout supplies x, y, and all later placement phases remain unchanged. These fixed task-specific rules use no privileged block position, flow decoding, or LeWM predictive evaluations.

Table 27: **Cube shared-stack control reference.** Counts use the same 1,000 TEST episodes; differences are height-informed rule minus unranked prior. 95% intervals use an episode-paired normal approximation from discordant pairs without continuity correction; p is two-sided exact McNemar. The rule has additional proprioceptive height. Neither arm is the headline Source-CEM configuration.

Outcome	Unranked prior	Height-informed rule	Difference [95% CI]	Exact p
Native 4-cm reach	853	928	$+7.5$ pp $[5.5, 9.5]$	3.28×10^{-14}
Semantic completion	848	925	$+7.7$ pp $[5.7, 9.7]$	1.11×10^{-14}

The T1 semantic-completion rule requires native reach, completed release, and containment at the post-action observation following the third opening command. With additional proprioceptive height and the shared execution stack, simple rules also achieve high Cube success. Both arms still use the frozen JEPA encoder through their patch-token readouts.

G.4 FULL-PANEL BOUNDED-RECOVERY INTERVENTION

Table 28 compares Recovery ON and OFF on the same 1,000-episode panel with identical frozen models, nominal configuration, and event-triggered perturbation rule; only the complete bounded recovery bundle changes. All 746 semantic-completion discordances favor ON. The 58 OFF completions also occur under ON, and all lie in the respective uninduced sets.

ON can recover from natural contact loss before the forced perturbation, so trajectories need not remain step-wise matched. The full-panel paired comparison estimates the bundle's effect; the matched-event subset below describes execution mechanisms.

Semantic completion is primary because 326 induced episodes first reach the 4-cm target before sustained contact loss is confirmed in both arms. A further 450 ON and 10 OFF episodes first reach at or after loss confirmation; the remaining 61 native reaches are identical uninduced episodes. The

Table 28: **Cube T3 recovery ON/OFF.** Outcomes use all 1,000 paired episodes. Differences are ON minus OFF; p is the exact McNemar test on episode outcomes.

Outcome	Recovery ON	Recovery OFF	Difference	McNemar p
Semantic completion	804/1000 (80.4%)	58/1000 (5.8%)	+74.6 pp	5.4×10^{-225}
Native 4-cm reach	837/1000 (83.7%)	397/1000 (39.7%)	+44.0 pp	7.04×10^{-133}

recovery-specific reach diagnostic below instead requires target attainment after re-entry into CARRY following the induced loss, regardless of earlier target entry.

G.5 MATCHED-EVENT RECOVERY AND ADDITIONAL DIAGNOSTICS

Of 839 ON episodes with induced sustained gripper-contact loss, 837 have an OFF counterpart with matching perturbation and loss-confirmation steps. Table 29 separates re-acquisition, recovery-specific reach, and completion in this matched-event subset.

Table 29: **Recovery on 837 matched-event episodes.** Both arms have identical perturbation and loss-confirmation steps. Recovery-specific reach requires post-loss re-entry into CARRY; semantic completion uses the usual post-release criterion.

Post-loss outcome	Recovery ON	Recovery OFF
Re-entry into CARRY	803/837	12/837
Recovery-specific native reach	760/837	0/837
Semantic completion	746/837 (89.1%)	0/837

Among all 839 Recovery-ON episodes with induced loss, 833 establish a new contact segment; the three subsequent outcome counts are unchanged from Table 29. A separate physical-drop predicate using drop records and a 20-mm lift threshold identifies 678 episodes, of which 620 complete (91.4%), distinguishing induced contact loss from evaluator-confirmed physical drops.

Recovery OFF induces sustained gripper-contact loss in 843 episodes: 825 have no subsequent contact, 14 re-grasp but do not satisfy the recovery-specific reach criterion, four release without containment after the third opening command, and none completes. Of the 14 broader re-grasps, 12 re-enter CARRY after the induced loss without rollback, recovery assistance, or recovery budget; none completes.

H QUALITATIVE ROLLOUTS

Figure 8 complements the quantitative results in Sec. 6 with qualitative rollout sequences.

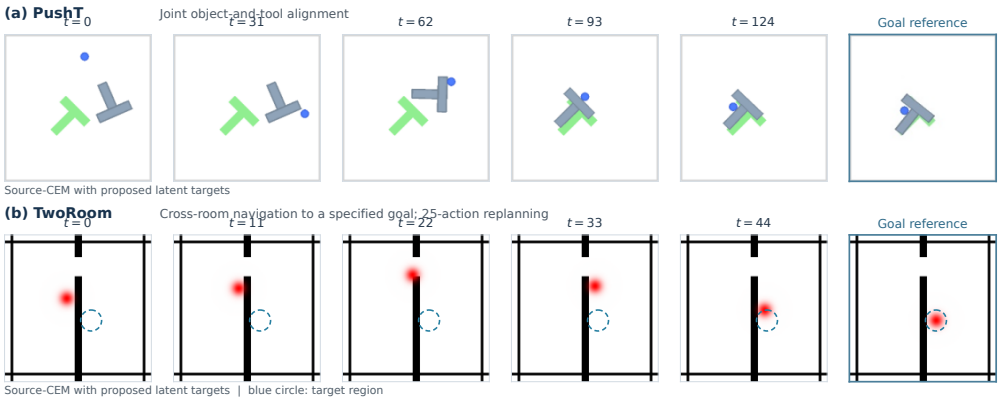

Figure 8: **Successful execution on PushT and TwoRoom.** (a) A pusher moves a T-shaped block to the target pose while also matching the target pusher position. (b) A disk-shaped agent crosses the doorway and reaches a specified position in the other room under collision-aware dynamics; the dashed blue circle marks the 16-pixel target region. Both sequences use CompassFlow Source-CEM with proposed latent targets, trajectory-field checkpoint 0, and 192 predictive evaluations per plan; replanning occurs every five actions on PushT and every 25 on TwoRoom. The five execution frames are evenly spaced between initialization and the first successful post-action observation; t counts executed actions, and the rightmost column shows the goal reference. These are illustrative reruns: PushT uses a non-TEST case and TwoRoom a strict TEST case; neither contributes to the reported rates.