LEARNING FROM LAST-MILE FAILURES IN FINE-GRAINED ROBOTIC MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Fine-grained robotic manipulation operates at tight tolerances; therefore, real-robot data collection inevitably yields many failed trials. Standard practice discards these failures or compresses each into a scalar reward, preference, or advantage. We take a global view: all trials form one record of how a task is performed, and success is a label set by the task tolerance. This record reveals a sharp regularity: most failures are **last-mile**, reproducing the successful motion and departing only where contact decides the outcome. We show that this structure turns a trajectory-level outcome label into a precise correction signal. (i) Every two-weight guidance rule over success-, failure-, and null-conditioned predictions, including existing attraction, repulsion, and two-branch rules, acts along one contrast direction and differs only in its gain. (ii) This direction aligns with the task-critical correction whenever a measurable scaffold ratio is below one, as for last-mile failures, so one label per trajectory localizes the error without step annotations. On this basis, we propose **CALM** (**C**ontrast **A**t the **L**ast **M**ile), which trains a world action model with outcome tokens and applies contrastive denoising guidance that attracts actions toward success and repels them from failure. On six real-robot tasks, CALM improves average success by 12.8 points over the standard baseline, with gains on every task, and loses this gain once off-scaffold failures raise the scaffold ratio above one, as the validity condition predicts.

1 INTRODUCTION

Fine-grained manipulation, such as insertion, slotting, and stacking, is decided within millimeters and degrees at the moment of contact. Robot policies for such tasks learn from teleoperated demonstrations collected on real robots (Chi et al., 2025; Black et al., 2025a; Bu et al., 2025; Kim et al., 2026), and at these tolerances a large share of the collected trials fail (Lu et al., 2025; Lin et al., 2026; Physical Intelligence et al., 2025). Standard practice treats these failures as waste: it trains on successes alone, or compresses each failure into a scalar reward, a preference, or an advantage (Kostrikov et al., 2022; Rafailov et al., 2023; Frans et al., 2025), a single number that penalizes the correct part of a failed trial along with its error. We take a global view of the data: the collected trials, successful or not, form one record of how the task is performed, and success is a label set by the task tolerance. A peg that misses its hole by a millimeter fails, yet nearly all of its motion is correct. This observation raises the central question of this work: **when does a failed trial, labeled only by its outcome, carry the correction that a successful execution needs?**

To answer this question, we conduct a systematic study of failure data in fine-grained manipulation. We represent each task by its *success scaffold*, the motion shared by successful executions together with their tolerances, and measure how long each failed trial stays on it before release. A striking regularity emerges (Fig. 1a,b): failures reproduce the approach and transport of successful executions and diverge only within the short window in which contact determines the outcome. We refer to them as **last-mile failures**. The regularity holds across data sources: it characterizes 96.2% of the procedure-complete failures of human teleoperation, holds across six operators, and reappears in public teleoperation data (Sliwowski et al., 2025).

This structure changes what a failure label can teach. Contrasting a success with a failure that shares its scaffold cancels everything the two have in common and leaves only the residual that decided the outcome (Fig. 1c). We formalize this intuition with outcome-conditioned denoising, where the

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.

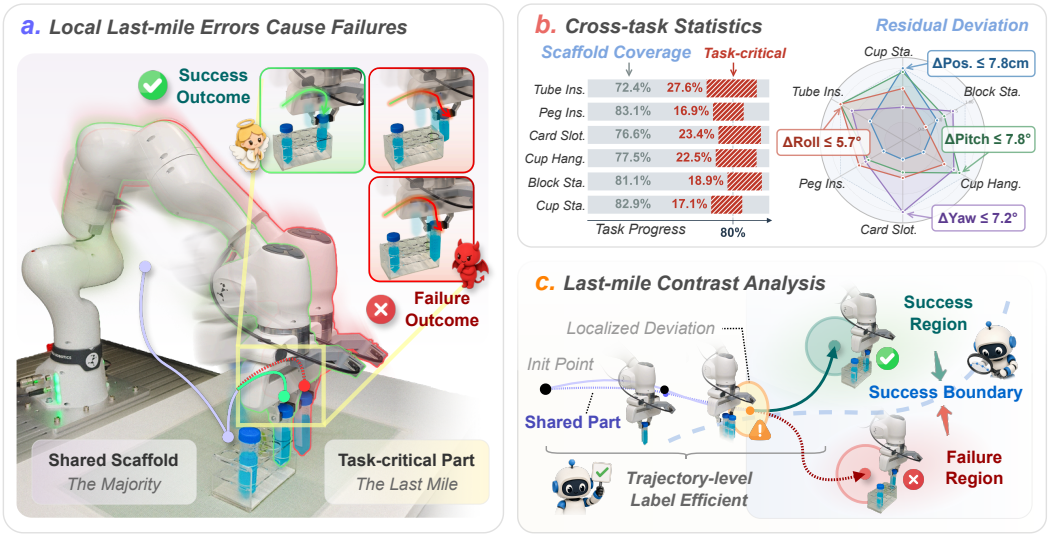


Figure 1: **Last-mile failures.** (a) A last-mile failure leaves the success scaffold only near contact. (b) Per task, the share of the pre-retract window that teleoperated failures spend on the scaffold, and their largest residuals. (c) The success–failure contrast cancels the scaffold, leaving the correction.

success–failure score difference is the gradient of the outcome log-odds, and establish two results. (i) **One contrast direction.** Every two-weight guidance rule over success, failure, and null predictions, including the attraction-only, repulsion-only, and two-branch rules of prior work (Frans et al., 2025; Ban et al., 2024; Liang et al., 2026), moves samples along this single direction; rules differ only in a state-dependent gain, while the data set the direction. (ii) **A validity condition.** Under a local residual model, the direction aligns with the task-critical correction whenever a measurable *scaffold ratio* is below one. On our tasks, last-mile failures satisfy this condition, so one trajectory-level label localizes the correction with no step annotations or critic.

These results also suggest where failure-aware learning should live. World action models (WAMs), which predict future observations together with actions on video-pretrained backbones, capture fine-grained robot-object interactions and physical dynamics and have become strong real-robot policies (Zhu et al., 2025b; Ye et al., 2026; Kim et al., 2026). They are a natural home for last-mile failures: a failure differs from a success by millimeters in action space, yet its consequence, a jammed peg or a collapsed stack, is plain to see in the predicted future. We therefore propose **CALM** (**C**ontrast **A**t the **L**ast **M**ile), which turns binary outcome labels into denoising-time correction inside a WAM. (i) **Outcome-conditioned training** (OCT) trains a single WAM denoiser under success, failure, and null outcome tokens, so each label conditions both the action chunk and its predicted consequence. (ii) **Contrastive denoising guidance** (CDG) composes these predictions at every denoising step, attracting samples toward success and repelling them from failure along the contrast direction above, with no critic or step annotations.

Across six fine-grained real-robot tasks, CALM improves average success by 12.8 points over the strongest of 6 baselines that discard, imitate, or rank failures, with gains on every task, and outperforms a separately trained outcome classifier on the same backbone by 8.9 points. The gain disappears where the validity condition fails: replacing half of the teleoperated failures with off-scaffold executions raises the scaffold ratio to 1.08 and lowers success to the level of attraction-only guidance, and the audit flags such pools before any training. Our contributions are threefold:

- **Empirical insight.** We audit failure data in fine-grained manipulation before training and find it largely last-mile: failures leave the successful motion only where contact decides the outcome.
- **Theoretical analysis.** We unify existing two-weight guidance rules as gains along one contrast direction and derive a measurable condition under which it aligns with the task-critical correction.
- **Practical framework.** We propose CALM, which turns binary outcome labels into denoising-time correction in a WAM and improves over all standard baselines on six real-robot tasks.

2 Related Work

Guidance, outcome conditioning, and policy improvement. Classifier and classifier-free guidance steer diffusion sampling with a label log-likelihood gradient (Dhariwal & Nichol, 2021; Ho & Salimans, 2022). Summed conditional scores compose concepts by conjunction and negation (Liu et al., 2022), and negative prompting repels an unwanted condition with a constant, posterior-dependent, or contrastive weight (Ban et al., 2024; Koulischer et al., 2025; Chang et al., 2026). In decision making, returns guide or condition trajectories (Janner et al., 2022; Ajay et al., 2023), and CFGRL shows that guidance toward an optimality token performs policy improvement (Frans et al., 2025). RECAP scales advantage conditioning to vision-language-action models (Physical Intelligence et al., 2025). DIPOLE trains reward-maximizing and reward-minimizing diffusion policies with return-derived weights and combines their predictions linearly (Liang et al., 2026). These methods differ in how they obtain the weights; Prop. 1 places all of them in one family, and Prop. 2 states when a terminal failure label, without returns or advantages, makes their shared direction corrective.

Learning from failed and imperfect robot data. Offline reinforcement learning fits values to mixed data (Kostrikov et al., 2022; Kumar et al., 2020; Nakamoto et al., 2023), preference methods compare or rate outcomes (Rafailov et al., 2023; Zhang et al., 2024; Ethayarajh et al., 2024; Li et al., 2024), and imitation from imperfect demonstrations down-weights or contrasts undesirable data (Jang et al., 2023; Hoang et al., 2025). Each reduces a failure to a value, a preference, or a weight that must then be assigned to its steps, by a critic or uniformly; in CALM, the steps a failure shares with successes cancel in the outcome contrast. Failed demonstrations can also supply useful segments (Wu et al., 2025) or be corrected interactively (Welte et al., 2026); CALM uses each failure whole, with one outcome label. In contrastive learning, negatives close to positives are most informative (Robinson et al., 2021; Jung et al., 2025). Last-mile failures are such negatives; we measure their frequency on real robots and when one outcome label yields a corrective direction.

Failures as generation-time guidance. Several works let failures act on generation directly. CCDP repels resampling from failed attempts within an episode (Razmjoo et al., 2025), and PPGuide steers a frozen policy with a learned success predictor (Wang et al., 2026). TraceFlow shows that one terminal bit per rollout can steer a frozen flow policy through retrieved success and failure traces (Zhang et al., 2026). Concurrently, AFIL trains separate success and failure action heads on a shared vision-language-action backbone and applies cosine-adaptive negative guidance with failures from online rollouts in simulation (Zheng et al., 2026). CARF repels failure-critical segments found by a progress scorer (Zhao et al., 2026b); other preprints shape flow directions with progress labels from simulated rollouts (Chen et al., 2026) or supervise world action models with failures (Peng et al., 2026). CALM learns the contrast inside one outcome-token denoiser from offline, naturally occurring real-robot failures, and Prop. 2 states when repulsion from failures corrects (App. F).

Robot policies and world action models. Diffusion and flow policies (Chi et al., 2025; Reuss et al., 2023) and vision-language-action models (Kim et al., 2024; Octo Model Team et al., 2024; Black et al., 2025b;a) learn multimodal visuomotor skills from demonstrations. Video-pretrained policies (Du et al., 2023b; Wu et al., 2024a; Hu et al., 2025) and world action models, which predict actions together with future observations (Zhu et al., 2025a; Li et al., 2025a; Ye et al., 2026; Kim et al., 2026), bring priors from video generation to control. CALM builds on the world action model of Cosmos Policy: it keeps the observation context fixed, guides only along the outcome token, and lets the contrast act on each action chunk together with its predicted outcome.

3 Last-Mile Failures and Why They Help

In fine-grained manipulation, contact decides the outcome within millimeters, so a failed trial usually repeats the approach and transport of a success and misses only at the end. We first measure this structure on real robots and then show why it lets a single outcome label teach the correction.

3.1 Last-Mile Failures in Fine-Grained Manipulation

The audit. A single audit measures this structure, with the same rules for every failure source (App. B). For each task, we align trajectories at grasp and release and fit the *success scaffold*, the per-step mean end-effector pose of the successes, with a *success envelope* of per-step tolerances around it. The outcome is decided before release, so the audit scores the K_{t_2} steps before it, the *pre-retract window*. A trajectory that first leaves the envelope for $W{=}10$ consecutive steps at step

k_{div} (k_{div}=K_{t_2} if it never does) has *pre-retract coverage*

$$C_{\text{int}} = k_{\text{div}}/K_{t_2}, \tag{1}$$

the share of the window before it first leaves the scaffold. A failure is *last-mile* when it completes the task procedure, as decided by fixed event rules and never by the metric, and has $C_{\text{int}} \geq 0.6$. On our tasks, the envelope uses task-specification tolerances fixed before any failure was inspected; they are loose during the approach and tighten at contact, to a few millimeters and degrees on the tightest tasks (Tab. 8). Multi-stage tasks are audited stage by stage, with one scaffold and one envelope per stage, since every stage of a long task has its own last mile.

Teleoperated failures are last-mile. Of our 132 procedure-complete teleoperated failures, 127 are last-mile, with at most one exception per task (Tab. 7). Per task, they stay on the scaffold for 72.4–83.1% of the pre-retract window on average (Fig. 1b): they repeat the approach and transport of a success and leave the scaffold only near contact. Counting the 20 screened-out attempts as not last-mile still leaves 127/152 (83.6%), and scaling every tolerance by $\times 0.5$ or $\times 1.5$ moves the count only to 123 or 130 of 132 (App. B.2). The pattern also holds at scale: the same audit, applied to more than 3K teleoperated trajectories that six operators collected for 28 short- and long-horizon tasks, finds about 82% of their failures last-mile across operators and tasks.

Public teleoperation agrees. On the public real-robot REASSEMBLE corpus (Sliwowski et al., 2025), with envelopes fit to successes alone, 147 of 212 failed insertions pass the audit, nearly as often as held-out successes (587/811; conformance ratio 0.96). Successful insertions into the neighboring hole pass in only 94 of 813 cases, so the audit detects a one-hole error (App. B.4).

Off-scaffold executions look different. As a stress set, we construct off-scaffold executions, 10 per task, by replaying held-out successes with a fixed offset added after grasp closure; their label is fixed by construction, never by the audit (App. B.3). They lie a median of 3.80 tolerance units from the scaffold, against 1.07 for our procedure-complete failures, and cover only 41.9% of the pre-retract window, about half the coverage of natural failures at every tolerance scale we test (App. B.2). Last-mile failures are thus the naturally occurring hard negatives of manipulation (Robinson et al., 2021): close to success yet on its wrong side, consistent with reports that VLM judges barely separate fine-grained success from failure (Sudhakar et al., 2026; Navasardyan et al., 2026). Near the success boundary, where such failures sit, an outcome label is also most informative: in a held-out logistic fit on the distance to the boundary, the normalized outcome variance falls from 1.00 near the boundary to 0.14 far from it, close to what the fitted logistic model predicts (App. E.4).

3.2 One Contrast Direction

At noise level σ, a denoiser trained with success, failure, and null outcome tokens has three branches $b \in \{+, -, \varnothing\}$ with scores $s_b = \nabla_x \log p^b_\sigma$, where the null branch models both outcomes. By Bayes' rule (Dhariwal & Nichol, 2021; Ho & Salimans, 2022), the success–failure difference is the gradient of the outcome log-odds for any class prior (App. A.1):

$$g_\sigma(x) \triangleq s_+(x) - s_-(x) = \nabla_x \log \frac{p_\sigma(c^+ \mid x)}{p_\sigma(c^- \mid x)}. \tag{2}$$

A trajectory label therefore always gives a direction, and the question is whether it points at the task-critical correction. When failures share the scaffold, the scaffold enters both densities in the same way and cancels in g_σ, leaving the last-mile residual (Fig. 1c). Two results make this precise, and each yields a quantity that we measure on the trained model.

Proposition 1 (One direction)**.** *For any weights $\lambda_\pm$, constant or state-dependent, the two-weight rule $\hat{s} = s_\varnothing + \lambda_+(s_+ - s_\varnothing) - \lambda_-(s_- - s_\varnothing)$ satisfies, with $q_\sigma(x) = p_\sigma(c^+ \mid x)$,*

$$\hat{s} = s_\varnothing + \kappa(x)\, g_\sigma, \qquad \kappa(x) = \lambda_+\big(1 - q_\sigma(x)\big) + \lambda_-\, q_\sigma(x). \tag{3}$$

Proposition 1 (proof in App. A.2) separates the two roles in every such rule: the rule sets only the gain, λ_+ on failure-like and λ_- on success-like states, while the data set the direction. Attraction-only, repulsion-only, and two-branch rules of prior work (Frans et al., 2025; Ban et al., 2024; Liang et al., 2026; Zheng et al., 2026) are all members of this family. On the null-free line $\lambda_+ = 1 + \lambda_-$ the null branch drops out, $\hat{s} = s_+ + \lambda_- g_\sigma$, so CALM's default $(2, 1)$ needs only the success and failure branches and does not depend on the training class ratio (App. A.2).

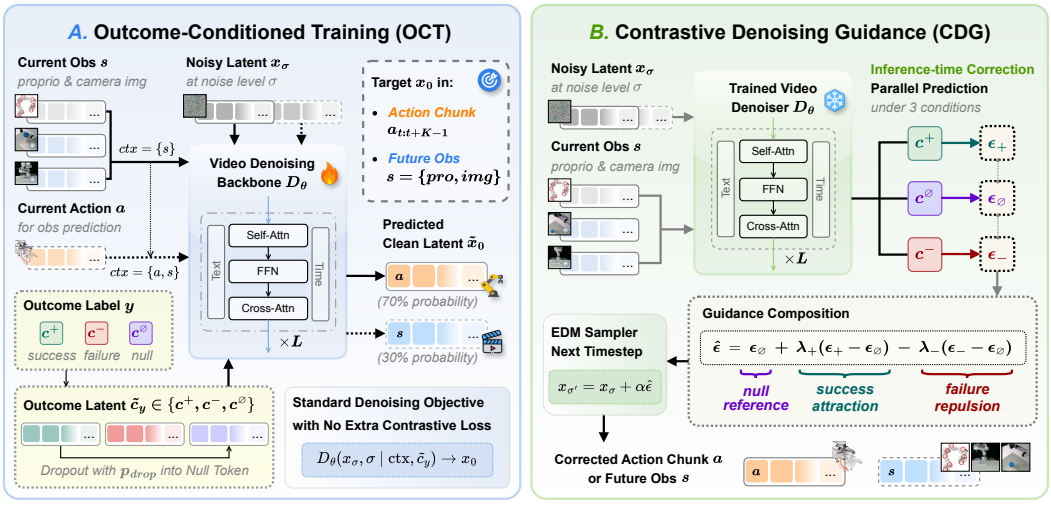


Figure 2: **Overview of CALM.** (A) OCT trains the world action model to denoise actions and future observations under success, failure, and null tokens, with no extra loss. (B) CDG composes the branches at inference to attract generation to success and repel it from last-mile failures.

3.3 When the Direction Corrects

The scaffold decides whether this direction points at the task-critical correction.

Proposition 2 (Alignment under a shared scaffold)**.** *Split $x = (x^{\text{sc}}, x^{\text{tc}})$ into scaffold and task-critical coordinates, with $g_\sigma = (g^{\text{sc}}, g^{\text{tc}})$. If $\|g^{\text{sc}}\| \le \delta$ and the task-critical residual is locally Gaussian with shared covariance C_σ and outcome means $\mu^\pm$ (Assumption 1), then the contrast g_σ aligns with the task-critical correction $u \triangleq C_\sigma^{-1}(\mu^+ - \mu^-)$:*

$$\cos \angle\big(g_\sigma, (0, u)\big) \;\ge\; \big(1 + \rho^2\big)^{-1/2}, \qquad \rho \;\triangleq\; \delta / \|u\|. \tag{4}$$

Under this local model, the contrast lies within $45°$ of the task-critical correction whenever the *scaffold ratio* ρ is below one, which we call the *alignment condition*. By Prop. 3, δ has two sources: a mismatch between the success and failure scaffolds, which the audit tracks before training, and a coupling of the task-critical residual to the scaffold state. Last-mile failures keep the first small, whereas off-scaffold failures inflate it and turn the contrast toward motion differences unrelated to the outcome. We therefore also measure $\hat{\rho}$ on each trained model from its own contrast $\hat{g} = (\epsilon_- - \epsilon_+)/\sigma$ (App. C.2); App. A gives the proofs and extends the bound to learned fields and to multi-stage tasks. The local model is checked on the learned field that guidance uses: a local reference fitted on a held-out calibration split agrees with it within relative error 0.18, 0.24, and 0.43 at low, mid, and high noise, so it describes the low-noise steps that set the final action (App. E.4).

Predictions. Four predictions follow; Tab. 14 lists them with the audit and boundary tests. (1) Both gains matter, and outcome guidance has no effect without failures, where $q_\sigma \equiv 1$ (Sec. 5.2, 5.3). (2) Natural failure pools satisfy $\hat{\rho} < 1$ (Sec. 5.4). (3) Replacing last-mile failures by off-scaffold executions raises $\hat{\rho}$ and lowers success (Sec. 5.4). (4) Under a shared scaffold, one label per trajectory recovers most of the gain of step labels without extra annotation (Sec. 5.5).

4 CALM: Contrast at the Last Mile

CALM instantiates this analysis in a world action model: one outcome-conditioned denoiser, trained on binary trajectory labels and sampled along the contrast of Prop. 1 (Fig. 2).

Why a world action model. Last-mile failures differ from successes by millimeters and degrees in action space, but their consequences are plain to see: the peg jams, the card misses the slot, the stack falls. A world action model (WAM) (Ye et al., 2026; Kim et al., 2026) trains one denoiser on the action chunk and the future observation in one latent sequence, so the outcome token conditions both targets: each failure label is tied to its visible consequence, and the contrast on actions comes

from the network that also predicts their outcomes. The future-observation target serves mainly the contrast, and video pretraining (Agarwal et al., 2025) adds a prior over contact dynamics.

Outcome-conditioned training. OCT trains the WAM of Cosmos Policy (Kim et al., 2026), fine-tuned from Cosmos-Predict2-2B (Agarwal et al., 2025), with three outcome tokens. Each chunk inherits the label $y \in \{+, -\}$ of its trajectory, and its token c^y is replaced by a `[NULL]` token $c^\varnothing$ with probability p_{drop}=0.15 (Ho & Salimans, 2022). Under this condition $\tilde{c}_y$, one denoiser D_θ minimizes the EDM loss (Karras et al., 2022) on all trajectories:

$$\mathcal{L}_{\text{OCT}}(\theta) \;=\; \mathbb{E}_{(s,a,s',y),\,\sigma,\,n}\Big[w(\sigma)\,\big\|M \odot \big(D_\theta(x_0 + \sigma n, \sigma \mid \text{ctx}, \tilde{c}_y) - x_0\big)\big\|_2^2\Big], \tag{5}$$

with noise $n \sim \mathcal{N}(0, I)$, EDM weighting $w(\sigma)$, and target mask M. Targets are routed 70/30 between the action chunk (x_0=a, ctx=s) and the future observation (x_0=s', ctx=(s, a)), with 2:1 success:failure mini-batches. No critic, reward model, or auxiliary loss is trained (App. C).

Contrastive denoising guidance. At each denoising step, CDG combines the branch predictions $\epsilon_b = (x_\sigma - D_\theta(x_\sigma, \sigma \mid \text{ctx}, c^b))/\sigma$, $b \in \{+, -, \varnothing\}$, into

$$\hat{\epsilon} \;=\; \epsilon_\varnothing + \lambda_+(\epsilon_+ - \epsilon_\varnothing) - \lambda_-(\epsilon_- - \epsilon_\varnothing), \tag{6}$$

the rule of Prop. 1 in ϵ form; its weights sum to one, so it reads the same in score, x_0, and velocity form (App. A.1). The weight λ_+ attracts a sample toward success and λ_- repels it from failure, both along the contrast direction of Prop. 1. CALM uses $(\lambda_+, \lambda_-) = (2, 1)$, which gives both gains and lies on the null-free line: the null coefficient $1 - \lambda_+ + \lambda_-$ is zero, so only the success and failure branches are evaluated, and `[NULL]` serves only as dropout in training.

Recipe. (1) *Audit* the failure pool (Sec. 3). (2) *Train* OCT on successes and failures with outcome tokens. (3) *Sample* with CDG at $(2, 1)$. The audit runs before any training and flags pools whose failures leave the scaffold, which inflates the scaffold ratio (Sec. 3.3). An optional gate built on it keeps 96% of natural failures and rejects 93% of constructed off-scaffold executions (App. E.3).

5 Experiments

5.1 Setup

Robot, tasks, and data. A Franka Research 3 arm with a wrist D405 and a front D435 camera performs six fine-grained tasks (App. E.1). They form three categories: *pose* (Tube and Peg Insertion), *orientation* (Card Slotting and Cup Hanging), and *stability* (Block and Cup Stacking). The ten training failures per task are drawn from the procedure-complete teleoperated failures of Sec. 3 without any coverage or distance threshold; the event rules that define this pool never read the audit metric. Every method of Tab. 1 sees 50 trajectories per task: 50 successes (50S) if it discards failures, and 40 successes plus 10 failures (40S+10F) if it uses them.

Real-time inference. CALM samples with 5 EDM steps and replans at 5 Hz. With all three branches batched, one action call takes 78 ms on an RTX 4090 (58 ms for one branch), an upper bound for the default, and the frame-to-command latency of 102 ms fits the 200 ms budget (Tab. 23).

Evaluation and statistics. Each method runs 30 rollouts per task from 30 initial configurations shared by all methods, 180 rollouts in total. Success rates pool these rollouts, and their 95% intervals are Wilson intervals. The primary comparison, CALM against the strongest standard baseline, reports a Newcombe 95% interval of the difference and a two-proportion z-test; since all methods share the initializations, we also report the largest exact McNemar p over every pairing of outcomes that the counts allow. All differences are in percentage points.

Baselines. They cover three ways to treat failures (App. C). (1) *Discard*: Cosmos Policy (Kim et al., 2026), whose backbone CALM shares, and $\pi_{0.5}$ (Black et al., 2025a), both on 50S. (2) *Imitate*: both policies on the pooled 40S+10F. (3) *Rank*: Cosmos Policy with Best-of-N selection over five candidates, and $\pi_{0.5}$ with DPO on success–failure pairs (Rafailov et al., 2023; Zhang et al., 2024). Four Cosmos Policy controls isolate the ingredients of CALM (Tab. 2). An outcome classifier, trained on noised latents of the same 40S+10F, guides a Cosmos Policy denoiser of the pooled data with the guidance potential of CALM (Dhariwal & Nichol, 2021), so it follows the rule of CALM at the population optimum and differs in how it estimates the contrast. With a binary terminal reward, this classifier estimates the Monte Carlo value of the data, the quantity that value-guided diffusion planning uses (Janner et al., 2022), so it represents critic-based use of the same failures. Three further controls use no failures: Cosmos Policy on 40S, a degenerate outcome-CFG that

Table 1: **Real-robot success rates (%).** Each task: 30 rollouts from matched initializations; Avg.: all 180. Success-only rows train on 50 successes per task (50S), the others on 40 successes and 10 failures (40S+10F). Δ: CALM minus row, in points.

Method	Data	Pose		Orientation		Stability		Avg.	Δ
		Tube	Peg	Card	Cup-H	Block	Cup-S		
Success-only *failures discarded*									
Cosmos Policy	50S	50.0	56.7	30.0	40.0	70.0	70.0	52.8	+24.4
$\pi_{0.5}$	50S	60.0	66.7	36.7	50.0	70.0	76.7	60.0	+17.2
Mixed imitation *failures imitated as demonstrations*									
Cosmos Policy	40S+10F	36.7	46.7	23.3	26.7	56.7	63.3	42.2	+35.0
$\pi_{0.5}$	40S+10F	46.7	56.7	26.7	40.0	66.7	66.7	50.6	+26.6
Trajectory preference *failures rank or pair rollouts*									
Cosmos Policy + BoN	40S+10F	60.0	56.7	36.7	46.7	70.0	76.7	57.8	+19.4
$\pi_{0.5}$ + DPO	40S+10F	63.3	70.0	43.3	56.7	73.3	80.0	64.4	+12.8
CALM (ours)	40S+10F	**76.7**	**80.0**	**60.0**	**70.0**	**86.7**	**90.0**	**77.2**	*ref.*

Table 2: **Cosmos-backbone controls.** 180 rollouts per column under the protocol of Tab. 1; the success-only columns use no failures. Δ: CALM minus column, in points.

	Success-only data				**Same success and failure data**		
	Cosmos Policy	Cosmos Policy	Degenerate outcome-CFG	Context-CFG	Mixed imitation	Outcome classifier	**CALM** (ours)
Data	40S	50S	50S	50S	40S+10F	40S+10F	40S+10F
Succ. (%)	48.9	52.8	53.3	58.9	42.2	68.3	**77.2**
Δ	+28.3	+24.4	+23.9	+18.3	+35.0	+8.9	*ref.*

labels every trajectory a success, and Context-CFG, which applies classifier-free guidance to the task context without outcome labels, isolating generic guidance.

5.2 Main Results: Failures Help Through the Contrast

CALM improves success on all six tasks. CALM reaches 77.2% (95% CI [70.6, 82.7]) against 64.4% ([57.2, 71.1]) for the strongest standard baseline, $\pi_{0.5}$ + DPO (Tab. 1). The gain of 12.8 points has a 95% CI of [3.4, 21.9] (z-test $p = 0.008$; McNemar $p \leq 0.032$ under every pairing), and on each task CALM exceeds the strongest baseline of that task by 10.0 to 16.7 points. The gain holds in every category: CALM reaches 78.3%, 65.0%, and 88.3% on the pose, orientation, and stability tasks, against 66.7%, 50.0%, and 76.7% for DPO. It is largest on the orientation tasks, which fail by a small rotation at contact and are the hardest category for every method; Cosmos Policy reaches only 35.0% on them, against 65.0% for CALM on the same tasks.

Imitating failures hurts. At the same 50 trajectories per task, pooling failures with successes lowers success from 52.8% to 42.2% for Cosmos Policy and from 60.0% to 50.6% for $\pi_{0.5}$. A last-mile failure repeats the successful approach and goes wrong only in the task-critical window, so imitation copies the very error that decided the outcome, with no label to mark it.

Ranking and a separate critic trail CALM. With the same failures, Best-of-N reaches 57.8%, DPO 64.4%, and the outcome classifier 68.3% (Tabs. 1 and 2). Selection cannot supply a correction absent from all five candidates, and a preference pair reduces each failure to a single comparison. The classifier follows the guidance rule of CALM (Prop. 1) but estimates its direction with a separate network on noised latents. CALM reads the same direction from the two conditional branches of its own denoiser, trained on the same data, and scores 8.9 points higher with no extra network.

The gain needs failures and guidance. Read left to right, the success-only block of Tab. 2 is a ladder. Ten more successes per task add 3.9 points to Cosmos Policy, whereas ten failures used by CALM add 28.3 points to the same 40 successes. A degenerate outcome branch adds 0.5 points, the null effect that Prop. 1 predicts when every label is a success ($q \equiv 1$). Context-CFG adds 6.1 points and still trails CALM by 18.3 points (McNemar $p \leq 0.003$). Guidance without failures can only sharpen the success distribution, and failures imitated without labels act as misleading demonstrations; the two strongest columns, the classifier and CALM, combine both ingredients.

Failures make a small budget go further. Trained with 30 trajectories per task, CALM reaches 66.7%, above every baseline family trained with 50, whose two-backbone means reach at most 61.1% (App. E.2). A small budget covers the outcome boundary sparsely, and each last-mile failure adds a label near the boundary, where an outcome label carries the most information (Sec. 3.1).

The correction appears where failures departed. In rollouts, success-only policies complete the approach and commit to a slightly offset contact pose, while CALM follows the same approach and corrects the pose just before contact, or adjusts contact and support before release (App. E.1). Executed CALM trajectories keep 68% of pre-retract steps within tolerance of the success scaffold and lie a median of 0.90 tolerance units from it, against 36% and 1.80 for success-only Cosmos Policy (App. E.3). This matches Prop. 2: when the scaffold contrast is small, the contrast points along the task-critical correction and leaves the shared approach intact.

CALM rollouts show three recurring patterns. (1) *Precision at contact.* In the insertion tasks, the end effector stays aligned with the target geometry through entry, without early contact; in placement and stacking, the object is better centered and oriented at the final interaction. (2) *Corrections where failures depart.* CALM follows the shared approach of the successes and adds small adjustments only near the task-critical phase, such as final alignment before insertion, orientation matching at the slot or rack, and stabilization of contact before release, which is where Sec. 3 locates the departure of last-mile failures. (3) *Recovery near the boundary.* When a rollout reaches a slightly misaligned or unstable state, CALM often readjusts pose, contact, or support before committing to the final action, most visibly in the orientation and stability tasks. Fig. 4 shows rollout frames per task; more rollout visualizations can be found in the Anonymous Repository.

5.3 The Rule Sets the Gain

Both gains contribute. Tab. 15 (App. E.2) samples the same CALM checkpoints under the rules of Prop. 1, changing only (λ_+, λ_-), which isolates each rule from the training pipeline that introduced it. The success branch alone, $(1, 0)$, reaches 59.4%, against 48.9% for Cosmos Policy on the same 40 successes, and guidance adds 17.8 more points at the default $(2, 1)$. Dropping a gain costs 13.9 points (McNemar $p \leq 0.020$) for the attraction-only rule $(2, 0)$ of optimality-token guidance (Frans et al., 2025; Physical Intelligence et al., 2025) and 20.0 points for the repulsion-only rule $(0, 1)$ of negative guidance (Ban et al., 2024). The rules with both gains, balanced $(1, 1)$, strong attraction $(3, 1)$, and the two-branch rule with w=2, $(3, 2)$, reach 71.7%, 76.1%, and 72.2%, all above the 63.3% of the best single-gain rule. The gains act on different states (Prop. 1): attraction stops pushing once a sample looks successful, and repulsion alone leaves failure-like samples uncorrected.

The default sits on a broad plateau. Success stays within 5.5 points of $(2, 1)$ over a six-fold range of λ_+, and every setting of either sweep with $\lambda_- \in [0.5, 3]$ stays above the strongest standard baseline (64.4%; App. E.2). The two gains saturate differently. Even λ_+=6 keeps 71.7%, since a sample pushed across the boundary leaves the reach of λ_+, whereas λ_-=4 falls to 62.2%: by Prop. 1, λ_- keeps acting on success-like samples and sets where they settle.

The linear contrast works best. The linear contrast of Prop. 1 beats KL-regularized, hinge, and reverse-KL forms by 10.0 to 12.2 points; each replaces it by a penalty or a clipped term (App. E.2).

5.4 The Data Set the Direction

By Prop. 2, the failure pool decides whether the contrast points at the correction, so Tab. 3 changes only the pool and Tab. 4 contaminates it step by step. Each Δ in Tab. 3 is taken against the attraction-only rule of the default model, which uses no repulsion. A negative Δ therefore means that training and guiding with that pool does worse than attraction alone.

Table 3: **Failure source.** CALM on 40S+10F per task, with the failures drawn from each source. Δ: row minus the attraction-only rule $(2, 0)$ of the default model.

Failure source	Succ. (%)	Δ vs. att.-only
Screened teleop. (default)	**77.2**	+13.9
Constructed off-scaffold	61.1	−2.2
Default model, att.-only $(2, 0)$	63.3	*ref.*

Table 4: **Contamination.** Constructed off-scaffold executions replace part of the 60 training failures; six-task means.

Replaced	$\hat{\rho}$	cos	Succ. (%)
0%	0.41	0.90	77.2
10%	0.52	0.86	74.4
25%	0.71	0.79	68.9
50%	1.08	0.68	62.8

Off-scaffold executions erase the gain. Trained on constructed off-scaffold executions, which leave the success envelope early by design, CALM reaches 61.1%, below the strongest standard baseline (64.4%) and the attraction-only rule (63.3%). By Prop. 2, their contrast has a large scaffold component, which pushes samples along scaffold directions unrelated to the task-critical correction; the contamination study below measures it. Measured before training, the audit already orders the two pools as their gains do: 96.2% of the teleoperated failures are last-mile, whereas the constructed executions cover only 41.9% of the pre-retract window.

Contamination moves alignment and success together. The natural pool satisfies the alignment condition in both metrics we use: $\hat{\rho} = 0.41$ in the covariance metric, in which $\|u\|$ is the Mahalanobis separation of the outcome means, and at most 0.52 per task in native coordinates, where the learned contrast lies within 26.8° to 35.1° of the task-critical residual (App. E.4). Replacing 6, 15, or 30 of the 60 training failures by constructed executions raises $\hat{\rho}$ to 0.52, 0.71, and 1.08. Meanwhile, the cosine to the task-critical direction falls from 0.90 to 0.68 and success from 77.2% to 62.8%. Once $\hat{\rho}$ exceeds one, so that the scaffold component of the contrast outweighs its task-critical component, CALM falls to the level of attraction-only guidance on clean data. The optional audit gate separates the two effects of replacement: at 50% it discards most constructed executions and keeps nearly all natural ones, and success recovers from 62.8% to 73.3%, so most of the loss comes from the constructed executions themselves, not from having fewer natural failures (App. E.3).

5.5 One Label per Trajectory, over Short and Long Horizons

Trajectory labels nearly match step labels. Retrained with labels per segment (4 per trajectory) or per step, CALM gains only 1.7 and 2.8 points, within evaluation noise, at about 1 and 4 annotation hours (Tab. 5). Trajectory labels need no annotation and recover 89.7% of the step-label gain over Cosmos Policy. On the clean pool, 82% of the contrast energy lies on task-critical blocks and 9% on scaffold blocks (App. E.3), so a label on the whole trajectory acts where the outcome is decided.

Trajectory labels extend to six-stage tasks. Each policy learns the full six-stage trajectory with one label per trajectory and no stage information; every failure fails one stage and completes the other five. On two six-stage real tasks, CALM improves over Cosmos Policy on both, from 27/60 to 36/60 successes in total (Tab. 6). App. A.6 explains why: the trajectory contrast stays small on the stages that a failure shares with successes, so it acts mainly on the stage where the failure departs. On the matched simulation benchmarks of App. D, CALM also improves over Cosmos Policy.

Table 5: **Label granularity.** CALM retrained with finer outcome labels on the same data; time is the approximate total annotation effort. Δ: row minus trajectory labels.

Supervision	Labels	Time	Succ. (%)	Δ
Success only (Cosmos)	none	none	52.8	−24.4
Trajectory (default)	1	≈0 h	**77.2**	*ref.*
Segment	4	≈1 h	78.9	+1.7
Step-wise	per step	≈4 h	80.0	+2.8

Table 6: **Long horizon.** Successes on two six-stage tasks: Cosmos Policy on 50S, CALM on 40S+10F.

Task	Cosmos	CALM
Setting Table	14/30	**19/30**
Cleaning Up	13/30	**17/30**
Both	27/60	**36/60**

6 Conclusion and Limitations

Most failures in our fine-grained real-robot tasks are last-mile, departing from the success scaffold only where contact decides the outcome. We showed that every two-weight success–failure guidance rule, existing ones included, moves samples along one contrast direction, which under a local Gaussian model aligns with the task-critical correction whenever the scaffold ratio is below one, and that off-scaffold failures pushing it above one remove the gain. In a world action model, CALM learns this correction from one label per trajectory, with no critic or step labels. It improves over every standard baseline on six real-robot tasks and over Cosmos Policy on two six-stage tasks. The practical message is simple: keep the failed trials, audit them, and label each by its outcome.

Limitations. Our study uses failures collected offline, each labeled once by its outcome. Collecting failures online during deployment would let CALM improve continually, and locating the last mile of each stage would extend it to stage-wise corrections in long-horizon tasks; in both settings, the audit can screen new failure sources before training.

AI USE STATEMENT

In this work, we used generative AI tools to draft and edit text, search the literature, and check bibliography entries. We did not use generative AI tools to collect robot data, or to run robot experiments. The authors verified all AI-assisted content and take full responsibility for the paper.

REPRODUCIBILITY STATEMENT

Sec. 3 and App. B specify the last-mile audit, including the procedure screen and its fixed exclusion rules, for our corpus and REASSEMBLE. App. A states the assumptions and gives complete proofs of Props. 1 and 2. App. C lists hyperparameters and baseline implementations, and App. E gives the evaluation protocol and the additional real-robot results. Side-by-side videos of all six real-robot tasks, comparing CALM with a baseline, are included in the supplementary material.

REFERENCES

Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical AI. *arXiv preprint arXiv:2501.03575*, 2025.

Christopher Agia, Rohan Sinha, Jingyun Yang, Zi-ang Cao, Rika Antonova, Marco Pavone, and Jeannette Bohg. Unpacking failure modes of generative policies: Runtime monitoring of consistency and progress. In *Conference on Robot Learning*, pp. 689–723, 2024.

Christopher Agia, Rohan Sinha, Jingyun Yang, Rika Antonova, Marco Pavone, Haruki Nishimura, Masha Itkina, and Jeannette Bohg. CUPID: Curating data your robot loves with influence functions. In *Conference on Robot Learning*, 2025.

Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making? In *International Conference on Learning Representations*, 2023.

Mohammadreza Armandpour, Ali Sadeghian, Huangjie Zheng, Amir Sadeghian, and Mingyuan Zhou. Re-imagine the negative prompt algorithm: Transform 2D diffusion into 3D, alleviate Janus problem and beyond. *arXiv preprint arXiv:2304.04968*, 2023.

Yuanhao Ban, Ruochen Wang, Tianyi Zhou, Minhao Cheng, Boqing Gong, and Cho-Jui Hsieh. Understanding the impact of negative prompts: When and how do they take effect? In *European Conference on Computer Vision*, pp. 190–206. Springer, 2024.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y Galliker, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In *Conference on Robot Learning*, 2025a.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. In *Robotics: Science and Systems*, 2025b.

Arwen Bradley and Preetum Nakkiran. Classifier-free guidance is a predictor-corrector. *arXiv preprint arXiv:2408.09000*, 2024.

Daniel S. Brown, Wonjoon Goo, Prabhat Nagarajan, and Scott Niekum. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In *International Conference on Machine Learning*, pp. 783–792, 2019.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. UniVLA: Learning to act anywhere with task-centric latent actions. In *Robotics: Science and Systems*, 2025.

Jinho Chang, Changsun Lee, Hyungjin Chung, and Jong Chul Ye. ContrastiveCFG: Guiding diffusion sampling by contrasting positive and negative concepts. In *International Conference on Machine Learning*, 2026.

Annie S. Chen, Alec M. Lessing, Yuejiang Liu, and Chelsea Finn. Curating demonstrations using online experience. *arXiv preprint arXiv:2503.03707*, 2025a.

Yanzhe Chen, Zhijun Cao, and Mike Zheng Shou. Where success breaks: Failure-boundary learning for robust vision-language-action models. *arXiv preprint arXiv:2609.06114*, 2026.

Yuxin Chen, Devesh K Jha, Masayoshi Tomizuka, and Diego Romeres. FDPP: Fine-tune diffusion policy with human preference. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 12010–12016. IEEE, 2025b.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

Muthu Chidambaram, Khashayar Gatmiry, Sitan Chen, Holden Lee, and Jianfeng Lu. What does guidance do? A fine-grained analysis in a simple setting. In *Advances in Neural Information Processing Systems*, volume 37, pp. 84968–85005, 2024.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, pp. 8780–8794, 2021.

Yilun Du, Conor Durkan, Robin Strudel, Joshua B Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Sohl-Dickstein, Arnaud Doucet, and Will Sussman Grathwohl. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and MCMC. In *International Conference on Machine Learning*, pp. 8489–8510. PMLR, 2023a.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 9156–9172, 2023b.

Jiafei Duan, Wilbert Pumacay, Nishanth Kumar, Yi Ru Wang, Shulin Tian, Wentao Yuan, Ranjay Krishna, Dieter Fox, Ajay Mandlekar, and Yijie Guo. AHA: A vision-language-model for detecting and reasoning over failures in robotic manipulation. In *International Conference on Learning Representations*, 2025.

Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. KTO: Model alignment as prospect theoretic optimization. In *International Conference on Machine Learning*, 2024.

Ronald A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7 (2):179–188, 1936.

Kevin Frans, Seohong Park, Pieter Abbeel, and Sergey Levine. Diffusion guidance is a controllable policy improvement operator. *arXiv preprint arXiv:2505.23458*, 2025.

Daniel H. Grollman and Aude Billard. Donut as I do: Learning from failed demonstrations. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3804–3809, 2011.

Philippe Hansen-Estruch, Ilya Kostrikov, Michael Janner, Jakub Grudzien Kuba, and Sergey Levine. IDQL: Implicit Q-learning as an actor-critic method with diffusion policies. *arXiv preprint arXiv:2304.10573*, 2023.

Joey Hejna, Rafael Rafailov, Harshit Sikchi, Chelsea Finn, Scott Niekum, W. Bradley Knox, and Dorsa Sadigh. Contrastive preference learning: Learning from human feedback without RL. In *International Conference on Learning Representations*, 2024.

Joey Hejna, Suvir Mirchandani, Ashwin Balakrishna, Annie Xie, Ayzaan Wahid, Jonathan Tompson, Pannag Sanketi, Dhruv Shah, Coline Devin, and Dorsa Sadigh. Robot data curation with mutual information estimators. In *Robotics: Science and Systems*, 2025.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Huy Hoang, Tien Mai, Pradeep Varakantham, and Tanvi Verma. Learning what to do and what not to do: Offline imitation from expert and undesirable demonstrations. *arXiv preprint arXiv:2505.21182*, 2025.

Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. In *International Conference on Machine Learning*, 2025.

Youngsoo Jang, Geon-Hyeong Kim, Jongmin Lee, Sungryull Sohn, Byoungjip Kim, Honglak Lee, and Moontae Lee. SafeDICE: Offline safe imitation learning with non-preferred demonstrations. In *Advances in Neural Information Processing Systems*, volume 36, pp. 74921–74951, 2023.

Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning*, 2022.

Yeonsung Jung, Trilok Padhi, Sina Shaham, Dipika Khullar, Joonhyun Jeong, Ninareh Mehrabi, and Eunho Yang. Co-evolving agents: Learning from failures as hard negatives. *arXiv preprint arXiv:2511.22254*, 2025.

Aleksandra Kalinowska, Ahalya Prabhakar, Kathleen Fitzsimons, and Todd Murphey. Ergodic imitation: Learning from what to do and what not to do. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3648–3654. IEEE, 2021.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 26565–26577, 2022.

Michael Kelly, Chelsea Sidrane, Katherine Driggs-Campbell, and Mykel J. Kochenderfer. HG-DAgger: Interactive imitation learning with human experts. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8077–8083, 2019.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. DROID: A large-scale in-the-wild robot manipulation dataset. In *Robotics: Science and Systems*, 2024.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. OpenVLA: An open-source vision-language-action model. In *Conference on Robot Learning*, 2024.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, et al. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*, 2026.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit Q-learning. In *International Conference on Learning Representations*, 2022.

Felix Koulischer, Johannes Deleu, Gabriel Raya, Thomas Demeester, and Luca Ambrogioni. Dynamic negative guidance of diffusion models. In *International Conference on Learning Representations*, volume 2025, pp. 34892–34921, 2025.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1179–1191, 2020.

Krunoslav Lehman Pavasovic, Jakob Verbeek, Giulio Biroli, and Marc Mezard. Overshoot and shrinkage in classifier-free guidance: From theory to practice. In *International Conference on Learning Representations*, volume 2026, pp. 132722–132771, 2026.

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. *arXiv preprint arXiv:2503.00200*, 2025a.

Shufan Li, Konstantinos Kallidromitis, Akash Gokul, Yusuke Kato, and Kazuki Kozuka. Aligning diffusion models by optimizing human utility. In *Advances in Neural Information Processing Systems*, volume 37, pp. 24897–24925, 2024.

Xiang Li, Rongrong Wang, and Qing Qu. Towards understanding the mechanisms of classifier-free guidance. In *Advances in Neural Information Processing Systems*, 2025b.

Ruiming Liang, Yinan Zheng, Kexin Zheng, Tianyi Tan, Jianxiong Li, Liyuan Mao, Zhihao Wang, Guang Chen, Hangjun Ye, Jingjing Liu, et al. Dichotomous diffusion policy optimization. In *International Conference on Learning Representations*, volume 2026, pp. 61669–61690, 2026.

Zijun Lin, Jiafei Duan, Haoquan Fang, Dieter Fox, Ranjay Krishna, Cheston Tan, and Bihan Wen. FailSafe: Reasoning and recovery from failures in vision-language-action models. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2026.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 44776–44791, 2023a.

Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum. Compositional visual generation with composable diffusion models. In *European Conference on Computer Vision*, pp. 423–439. Springer, 2022.

Zeyi Liu, Arpit Bahety, and Shuran Song. REFLECT: Summarizing robot experiences for failure explanation and correction. In *Conference on Robot Learning*, 2023b.

Cheng Lu, Huayu Chen, Jianfei Chen, Hang Su, Chongxuan Li, and Jun Zhu. Contrastive energy prediction for exact energy-guided diffusion sampling in offline reinforcement learning. In *International Conference on Machine Learning*, pp. 22825–22855, 2023.

Guanxing Lu, Wenkai Guo, Chubin Zhang, Yuheng Zhou, Haonan Jiang, Zifeng Gao, Yansong Tang, and Ziwei Wang. VLA-RL: Towards masterful and general robotic manipulation with scalable reinforcement learning. *arXiv preprint arXiv:2505.18719*, 2025.

Jianlan Luo, Perry Dong, Yuexiang Zhai, Yi Ma, and Sergey Levine. RLIF: Interactive imitation learning as reinforcement learning. In *International Conference on Learning Representations*, 2024.

Jianlan Luo, Charles Xu, Jeffrey Wu, and Sergey Levine. Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning. *Science Robotics*, 10(105):eads5033, 2025.

David J. C. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604, 1992.

Ajay Mandlekar, Danfei Xu, Roberto Martín-Martín, Yuke Zhu, Li Fei-Fei, and Silvio Savarese. Human-in-the-loop imitation learning using remote teleoperation. *arXiv preprint arXiv:2012.06733*, 2020.

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *Conference on Robot Learning*, 2021.

David McAllister, Songwei Ge, Brent Yi, Chung Min Kim, Ethan Weber, Hongsuk Choi, Haiwen Feng, and Angjoo Kanazawa. Flow matching policy gradients. In *International Conference on Learning Representations*, 2026.

Mitsuhiko Nakamoto, Simon Zhai, Anikait Singh, Max Sobol Mark, Yi Ma, Chelsea Finn, Aviral Kumar, and Sergey Levine. Cal-QL: Calibrated offline RL pre-training for efficient online fine-tuning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 62244–62269, 2023.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. RoboCasa: Large-scale simulation of everyday tasks for generalist robots. In *Robotics: Science and Systems*, 2024.

Zaruhi Navasardyan, Tatul Danielyan, and Hrant Davtyan. FailBench: How reliable are VLMs at judging robot task success? *arXiv preprint arXiv:2609.03611*, 2026.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. In *Robotics: Science and Systems*, 2024.

Nana Yaw Owusu Ofori-Ampofo, Samira Ebrahimi Kahou, and Joseph Thekinen. REBOOT: From failure to recovery – a dataset and benchmark for precision assembly. *arXiv preprint arXiv:2609.22591*, 2026.

Quanquan Peng, Yutong Liang, Rui Yan, Nicklas Hansen, and Xiaolong Wang. FACT: Failure-aware causal training for world-action models. *arXiv preprint arXiv:2608.10232*, 2026.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

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

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pp. 53728–53741, 2023.

Amirreza Razmjoo, Sylvain Calinon, Michael Gienger, and Fan Zhang. CCDP: Composition of conditional diffusion policies with guided sampling. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 20036–20043. IEEE, 2025.

Allen Ren, Justin Lidard, Lars Ankile, Anthony Simeonov, Pulkit Agrawal, Anirudha Majumdar, Benjamin Burchfiel, Hongkai Dai, and Max Simchowitz. Diffusion policy policy optimization. In *International Conference on Learning Representations*, volume 2025, pp. 77288–77329, 2025.

Moritz Reuss, Maximilian Li, Xiaogang Jia, and Rudolf Lioutikov. Goal-conditioned imitation learning using score-based diffusion policies. In *Robotics: Science and Systems*, 2023.

Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. In *International Conference on Learning Representations*, 2021.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

Marco Saerens, Patrice Latinne, and Christine Decaestecker. Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure. *Neural Computation*, 14(1):21–41, 2002.

Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

Daniel Sliwowski, Shail Jadav, Sergej Stanovcici, Jedrzej Orbik, Johannes Heidersberger, and Dongheui Lee. Demonstrating REASSEMBLE: A multimodal dataset for contact-rich robotic assembly and disassembly. In *Robotics: Science and Systems*, 2025.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Sruthi Sudhakar, Junbang Liang, Sreehari Rammohan, Pavel Tokmakov, Richard Zemel, and Carl Vondrick. Robot critics that sweat the small stuff. *arXiv preprint arXiv:2606.21572*, 2026.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8228–8238. IEEE, 2024.

Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. In *International Conference on Learning Representations*, 2023.

Zixing Wang, Devesh K Jha, Ahmed H Qureshi, and Diego Romeres. PPGuide: Steering diffusion policies with performance predictive guidance. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

Edgar Welte, Yitian Shi, Rosa Wolf, Maximillian Gilles, and Rania Rayyes. FlowCorrect: Efficient interactive correction of generative flow policies for robotic manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2026.

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. In *International Conference on Learning Representations*, volume 2024, pp. 10641–10662, 2024a.

Kun Wu, Ning Liu, Zhen Zhao, Di Qiu, Jinming Li, Zhengping Che, Zhiyuan Xu, and Jian Tang. Learning from imperfect demonstrations with self-supervision for robotic manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 16899–16906. IEEE, 2025.

Yuchen Wu, Minshuo Chen, Zihao Li, Mengdi Wang, and Yuting Wei. Theoretical insights for diffusion guidance: A case study for Gaussian mixture models. *arXiv preprint arXiv:2403.01639*, 2024b.

Yueh-Hua Wu, Nontawat Charoenphakdee, Han Bao, Voot Tangkaratt, and Masashi Sugiyama. Imitation learning from imperfect demonstration. In *International Conference on Machine Learning*, pp. 6818–6827. PMLR, 2019.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

Jiaxuan Zhang, Ruizhe Liu, Yu Zhang, and Yanchao Yang. TraceFlow: Guiding frozen flow-matching robot policies with success and failure traces. *arXiv preprint arXiv:2609.20646*, 2026.

Songyuan Zhang, Zhangjie Cao, Dorsa Sadigh, and Yanan Sui. Confidence-aware imitation learning from demonstrations with varying optimality. In *Advances in Neural Information Processing Systems*, pp. 12340–12350, 2021.

Tonghe Zhang, Chao Yu, Sichang Su, and Yu Wang. ReinFlow: Fine-tuning flow matching policy with online reinforcement learning. In *Advances in Neural Information Processing Systems*, 2025.

Zijian Zhang, Kaiyuan Zheng, Zhaorun Chen, Joel Jang, Yi Li, Siwei Han, Chaoqi Wang, Mingyu Ding, Dieter Fox, and Huaxiu Yao. GRAPE: Generalizing robot policy via preference alignment. *arXiv preprint arXiv:2411.19309*, 2024.

Botong Zhao, Fang Yu, Tim Yu, Senhua Zhu, Xinyuan Chen, and Yue Lu. PAVE: Predictive alignment and value-guided evolution for world-action policies. *arXiv preprint arXiv:2608.30378*, 2026a.

Shuqi Zhao, Bang Du, Cheng-En Wu, Yichen Xie, Yixiao Wang, and Masayoshi Tomizuka. CARF: Contrastive attraction-repulsion of failure-guided flow matching. *arXiv preprint arXiv:2609.21982*, 2026b.

Meng Zheng, Samhita Marri, Anwesa Choudhuri, Benjamin Planche, Zhongpai Gao, Van Nguyen Nguyen, Terrence Chen, Girish Chowdhary, and Ziyan Wu. Failing forward: Adaptive failure-informed learning for vision-language-action models. *arXiv preprint arXiv:2605.08434*, 2026.

Yao Zhou, Hang Gao, Fengge Wu, Changwen Zheng, and Wenwen Qiang. Temporal GRPO: Beyond trajectory-level credit in vision-language-action reinforcement learning. *arXiv preprint arXiv:2608.13026*, 2026.

Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. *arXiv preprint arXiv:2504.02792*, 2025a.

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. IRASim: A fine-grained world model for robot manipulation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9834–9844, 2025b.

A Theory and Proofs

This appendix proves Props. 1 and 2 and extends them to learned fields, contaminated failure pools, label information near the outcome boundary, and multi-stage tasks.

A.1 Conventions and Standard Identities

All statements are local to a fixed context, noise level σ, and query x, where all densities are positive and differentiable. CALM uses EDM noising $x_\sigma = x_0 + \sigma n$ with $n \sim \mathcal{N}(0, I)$ (Karras et al., 2022). For the MMSE denoiser $D^\star$, Tweedie's formula gives $\nabla_x \log p_\sigma = (D^\star - x)/\sigma^2$ (Efron, 2011; Vincent, 2011), so $\epsilon^\star \triangleq (x - D^\star)/\sigma = -\sigma \nabla_x \log p_\sigma$ at the population optimum. **Parameterizations.** The maps from scores to ϵ, to x_0, and to flow-matching velocities are affine with a branch-independent offset, so a combination $\sum_b w_b(\cdot)_b$ with $\sum_b w_b = 1$ is the same rule in every form. The weights $(1 - \lambda_+ + \lambda_-, \lambda_+, -\lambda_-)$ of Eq. (6) sum to one. **Signs.** $\epsilon_+ - \epsilon_- = -\sigma g_\sigma$, and an Euler step of the EDM probability-flow ODE $\mathrm{d}x/\mathrm{d}\sigma = \hat{\epsilon}$ with decreasing σ moves along $+\hat{s}$; the diagnostic $\hat{g} = (\epsilon_- - \epsilon_+)/\sigma$ therefore estimates $+g_\sigma$.

Standard identities. Outcomes are binary in training, with priors $\pi_\pm$ ($\pi_+ = 2/3$ under 2:1 mini-batches), and outcome dropout makes the null branch model the mixture $p_\sigma^\varnothing = \pi_+ p_\sigma^+ + \pi_- p_\sigma^-$.

Lemma 1 (Guidance identity and prior shift (Saerens et al., 2002; Dhariwal & Nichol, 2021; Ho & Salimans, 2022))**.** *$s_\pm - s_\varnothing = \nabla_x \log p_\sigma(c^\pm \mid x)$, and $\log\big(p_\sigma(c^+ \mid x)/p_\sigma(c^- \mid x)\big) = \log(p_\sigma^+/p_\sigma^-) + \log(\pi_+/\pi_-)$. Hence g_σ is the gradient of the posterior log-odds for any priors, Eq. (2).*

Proof. Bayes' rule gives $p_\sigma(c^\pm \mid x) = \pi_\pm p_\sigma^\pm(x)/p_\sigma^\varnothing(x)$, whose prior terms vanish under $\nabla_x \log$. □

A calibrated noise-conditional outcome classifier trained under the same prior represents the same field: guidance with the potential $\lambda_+ \log q_\sigma - \lambda_- \log(1 - q_\sigma)$ adds $\kappa\, g_\sigma$ to $s_\varnothing$. At the population optimum it therefore equals CALM, and the two differ only in how they estimate g_σ.

A.2 One Direction, Two Gains

Proof of Prop. 1. Let $L = \log(p_\sigma^+/p_\sigma^-) + \log(\pi_+/\pi_-)$, so $q_\sigma = \mathrm{sig}(L)$ and $\nabla_x L = g_\sigma$ (Lemma 1). With $\mathrm{sig}' = \mathrm{sig}(1-\mathrm{sig})$, $s_+ - s_\varnothing = \nabla \log q_\sigma = (1-q_\sigma) g_\sigma$ and $s_- - s_\varnothing = \nabla \log(1-q_\sigma) = -q_\sigma g_\sigma$. Both hold pointwise, so for any weights, constant or state-dependent, $\hat{s} = s_\varnothing + \lambda_+(1 - q_\sigma) g_\sigma + \lambda_- q_\sigma g_\sigma = s_\varnothing + \kappa\, g_\sigma$, which is Eq. (3) with the gain κ of the proposition. □

For constant weights $\hat{s} = \nabla_x \log\big[p_\sigma^\varnothing\, q_\sigma^{\lambda_+} (1 - q_\sigma)^{-\lambda_-}\big]$. Attraction alone is the product form of classifier guidance and of guidance toward an optimality token (Dhariwal & Nichol, 2021; Frans et al., 2025). Repulsion alone is the negative-guidance form (Ban et al., 2024), whose posterior-dependent strength Koulischer et al. (2025) analyze. The gain κ lies between λ_- and λ_+, and at high noise, where $q_\sigma \approx \pi_+$, $\kappa \approx \pi_- \lambda_+ + \pi_+ \lambda_-$. As for any guidance rule, the guided scores at different σ are in general not the marginal scores of one diffused distribution (Chidambaram et al., 2024; Bradley & Nakkiran, 2024); the null-free Gaussian case of Corollary 1 is an exact exception.

Corollary 1 (Null-free line under a local Gaussian model)**.** *Fix the scaffold and let $p_\sigma^\pm(x^{\mathrm{tc}}) = \mathcal{N}(\mu^\pm, C_0 + \sigma^2 I)$, the EDM marginals of $\mathcal{N}(\mu^\pm, C_0)$, with exact scores. If $\lambda_+ = 1 + \lambda_-$, then for every σ, $\hat{s}_\sigma$ is the score of $\mathcal{N}(\mu^+ + \lambda_- \Delta\mu,\, C_0 + \sigma^2 I)$, so guided sampling returns $\mathcal{N}(\mu^+ + \lambda_- \Delta\mu, C_0)$ exactly. For other weights, $\hat{s}_\sigma$ tends to the score of a Gaussian with mean $\mu^+ + \lambda_- \Delta\mu$ where $q_\sigma \to 1$ and with mean $\mu^+ + (\lambda_+ - 1)\Delta\mu$ where $q_\sigma \to 0$.*

Proof. On the line, $\hat{s} = (1 + \lambda_-) s_+ - \lambda_- s_- = -C_\sigma^{-1}(x^{\mathrm{tc}} - \mu^+ - \lambda_- \Delta\mu)$ with $C_\sigma = C_0 + \sigma^2 I$. Off the line, $s_\varnothing = q_\sigma s_+ + (1 - q_\sigma) s_-$ and κ tend to (s_+, λ_-) where $q_\sigma \to 1$ and to (s_-, λ_+) where $q_\sigma \to 0$, and Eq. (3) then gives the stated means of the two regimes. □

The corollary explains the measured rules of App. E.2. Attraction alone, $(2, 0)$, pushes failure-like samples one displacement past μ^+ but stops at μ^+ once a sample looks successful, and repulsion alone, $(0, 1)$, leaves failure-like samples at μ^-. A failure-like sample pushed across the boundary

enters the success-like regime, so λ_- sets where samples settle and a large λ_+ limits itself; on the null-free line both regimes meet at $\mu^+ + \lambda_-\Delta\mu$, so the default $(2, 1)$ shifts the success residual one full residual displacement further away from the failures.

Two-sided failures and several success modes. Assumption 1(b) conditions the outcome means on the scaffold state, $\mu^{\pm}(x^{\mathrm{sc}})$. Failures that miss on both sides of the success residual across the data, and successes that approach from several sides, therefore fit the local model whenever the scaffold state selects the side, as when the approach pose sets the direction of a missed insertion; the contrast is then state-dependent and points to the success residual of the current mode. If failures surround the success residual at one scaffold state, with equal means and a wider covariance $C^- \succ C^+$, then $g^{\mathrm{tc}} = -\big((C^+)^{-1} - (C^-)^{-1}\big)(x^{\mathrm{tc}} - \mu^+)$ points from every nearby state back to μ^+, so the contrast still restores the success residual and no longer shifts it. On our tasks the mean failure residual lies 0.76 to 0.91 local standard deviations from the success residual (Tab. 21), so the displacement is systematic; the default moves the success residual by one such displacement, within the spread of successes, while λ_-=4 moves it four times as far and falls to 62.2% (Tab. 17).

A.3 Alignment under a Shared Scaffold

Split the latent into scaffold and task-critical coordinates, $x = (x^{\mathrm{sc}}, x^{\mathrm{tc}})$, with $g_\sigma = (g^{\mathrm{sc}}, g^{\mathrm{tc}})$.

Assumption 1 (Local residual model)**.** At the query (σ, x): (a) $\|g^{\mathrm{sc}}(x)\| \le \delta$; (b) $p_\sigma(x^{\mathrm{tc}} \mid x^{\mathrm{sc}}, c^{\pm}) = \mathcal{N}\big(\mu^{\pm}(x^{\mathrm{sc}}), C_\sigma\big)$ with $C_\sigma \succ 0$ and $\Delta\mu \triangleq \mu^+ - \mu^- \neq 0$.

Part (a) is where the failure data enter, and part (b) models the residual at one query and noise level; App. E measures both on the learned fields. Let $u \triangleq C_\sigma^{-1}\Delta\mu$, $\bar{u} \triangleq (0, u)$, and $\rho = \delta/\|u\|$. The identity $g^{\mathrm{tc}} = u$ is Fisher's discriminant direction (Fisher, 1936); Prop. 2 adds the condition under which the full contrast follows it, in a quantity we can measure.

Proof of Prop. 2. By the chain rule, $\log p^{\pm}_\sigma(x) = \log p_\sigma(x^{\mathrm{sc}} \mid c^{\pm}) + \log p_\sigma(x^{\mathrm{tc}} \mid x^{\mathrm{sc}}, c^{\pm})$, whose first term does not depend on x^{tc}, so $g^{\mathrm{tc}} = -C_\sigma^{-1}(x^{\mathrm{tc}} - \mu^+) + C_\sigma^{-1}(x^{\mathrm{tc}} - \mu^-) = u$. Then $\langle g_\sigma, \bar{u}\rangle = \|u\|^2$ and $\|g_\sigma\| \le \|u\|(1+\rho^2)^{1/2}$, giving Eq. (4); $\rho < 1$ gives an angle below 45°. □

Proposition 3 (What δ measures)**.** *Under Assumption 1(b), let $J^{\pm} = \partial\mu^{\pm}/\partial x^{\mathrm{sc}}$. Then*

$$g^{\mathrm{sc}} = \nabla_{x^{\mathrm{sc}}} \log \frac{p_\sigma(x^{\mathrm{sc}} \mid c^+)}{p_\sigma(x^{\mathrm{sc}} \mid c^-)} + (J^+)^\top C_\sigma^{-1}(x^{\mathrm{tc}} - \mu^+) - (J^-)^\top C_\sigma^{-1}(x^{\mathrm{tc}} - \mu^-).$$

If $J^+ = J^- = J$, then $g^{\mathrm{sc}} = \nabla_{x^{\mathrm{sc}}} \log\big(p_\sigma(x^{\mathrm{sc}} \mid c^+)/p_\sigma(x^{\mathrm{sc}} \mid c^-)\big) - J^\top u$ and $\rho \le \delta_{\mathrm{sc}}/\|u\| + \|J\|_{\mathrm{op}}$, where δ_{sc} bounds the norm of the first term.

Proof. Differentiate the chain-rule decomposition in the proof of Prop. 2 with respect to x^{sc} term by term, using $\nabla_z \log \mathcal{N}\big(y; \mu(z), C\big) = J(z)^\top C^{-1}\big(y - \mu(z)\big)$. □

The scaffold ratio thus has two sources: a mismatch between the success and failure scaffolds, which the audit of Sec. 3.1 tracks in trajectory space before training, and a coupling of the task-critical mean to the scaffold state, which the audit cannot see. Neither is assumed small; the diagnostics measure their combined effect, $\hat{\rho}$, directly on the learned field of each trained model.

Lemma 2 (Learned contrast and magnitude)**.** *A learned $\hat{g}$ with $\|\hat{g} - g_\sigma\| \le e < \|u\|$ satisfies* $\cos\angle(\hat{g}, \bar{u}) \ge (1 - e/\|u\|)/(\sqrt{1+\rho^2} + e/\|u\|)$. *If $C_\sigma \preceq \Sigma + \sigma^2 I$ for the clean residual covariance Σ, then $\|u\| \ge \|\Delta\mu\|/(\|\Sigma\|_{\mathrm{op}} + \sigma^2)$.*

Proof. With $\hat{g} = g_\sigma + r$, $\langle \hat{g}, \bar{u}\rangle \ge \|u\|^2 - e\|u\|$ and $\|\hat{g}\| \le \|u\|(1+\rho^2)^{1/2} + e$. For the magnitude, $\|C_\sigma^{-1}\Delta\mu\| \ge \|\Delta\mu\|/\|C_\sigma\|_{\mathrm{op}}$. □

The covariance condition holds with equality when the clean residual is independent of the clean scaffold given the outcome, and it holds for a jointly Gaussian scaffold and residual, since conditioning subtracts a positive semidefinite Schur complement. Errors shared by the two branches cancel in $\hat{g}$, and for a block-diagonal metric $M = \mathrm{diag}(M_{\mathrm{sc}}, M_{\mathrm{tc}}) \succ 0$ both bounds hold in $\langle a, b\rangle_M = a^\top M b$. **Unequal covariances.** If $p_\sigma(x^{\mathrm{tc}} \mid x^{\mathrm{sc}}, c^{\pm}) = \mathcal{N}(\mu^{\pm}, C^{\pm})$ with $\|(C^{\pm})^{-1} - C_\sigma^{-1}\|_{\mathrm{op}} \le \zeta$

and $\|x^{\mathrm{tc}} - \mu^{\pm}\| \le R$, then $\|g^{\mathrm{tc}} - u\| \le 2\zeta R$, and Lemma 2 holds with $e + 2\zeta R$ in place of e, so a small covariance gap costs little alignment in practice.

Remark 1 (What the diagnostics test). Let $\hat{\rho} = \|\hat{g}^{\mathrm{sc}}\|/\|u\|$ and $\hat{e} = \|\hat{g}^{\mathrm{tc}} - u\|/\|u\| < 1$. Bounding $\langle \hat{g}, \bar{u} \rangle$ and $\|\hat{g}\|$ on both sides, with $\hat{g}^{\mathrm{tc}} = u + r$ and $\|r\| = \hat{e}\|u\|$, gives

$$\frac{1-\hat{e}}{\sqrt{\hat{\rho}^2 + (1+\hat{e})^2}} \;\le\; \cos\angle(\hat{g}, \bar{u}) \;\le\; \frac{1+\hat{e}}{\sqrt{\hat{\rho}^2 + (1-\hat{e})^2}},$$

with both sides equal to $(1+\hat{\rho}^2)^{-1/2}$ at $\hat{e} = 0$. The measured cosine thus depends on $\hat{\rho}$, which tests Assumption 1(a), and on $\hat{e}$, which tests (b). Sec. 5 reports $\hat{\rho}$ and the cosine, which together bound $\hat{e}$ from below through this inequality: a cosine below $(1+\hat{\rho}^2)^{-1/2}$ can only come from a deviation of the learned task-critical block from the reference direction u, which combines the estimation error of the learned field with the error of the fitted reference.

A.4 Contaminated Failure Pools

Let the failure pool mix last-mile and off-scaffold executions, $p_\sigma^- = (1-\alpha)p_\sigma^{\mathrm{lm}} + \alpha p_\sigma^{\mathrm{off}}$. Then $s_- = (1-\omega)\, s_-^{\mathrm{lm}} + \omega\, s_-^{\mathrm{off}}$ with responsibility $\omega(x) = \alpha\, p_\sigma^{\mathrm{off}}(x)/p_\sigma^-(x)$, so $g_\sigma - g_\sigma^{\mathrm{lm}} = -\omega\,(s_-^{\mathrm{off}} - s_-^{\mathrm{lm}})$ vanishes wherever ω does, that is, away from the off-scaffold executions.

Remark 2 (Contamination raises ρ and dilutes u). (1) Off-scaffold executions leave the envelope early, so their score departs from s_-^{lm} in the scaffold block; where ω is large, this increases δ and ρ. (2) If both components satisfy Assumption 1(b) with the same C_σ, then $g^{\mathrm{tc}} = (1-\omega)\, u^{\mathrm{lm}} + \omega\, u^{\mathrm{off}}$ with $u^{\mathrm{lm}} = C_\sigma^{-1}(\mu^+ - \mu^{\mathrm{lm}})$ and $u^{\mathrm{off}} = C_\sigma^{-1}(\mu^+ - \mu^{\mathrm{off}})$, so the last-mile direction keeps weight $1-\omega$. (3) The null branch also contains the failure pool, so by Prop. 1 contamination changes the attraction term $(1-q_\sigma)g_\sigma$ as well, although attraction alone never queries the failure branch.

A.5 Boundary Information

The Fisher information of a logistic label, the classical argument behind uncertainty sampling (MacKay, 1992; Settles, 2009), explains why labels near the success boundary carry the information that the contrast uses. Let $d_\theta(z) = d_0(z) - \theta$ be the signed distance of a trajectory embedding z to a local outcome boundary with offset θ, negative on the success side, and $P(c^+ \mid z; \theta) = \mathrm{sig}(-\beta d_\theta(z))$ with sharpness $\beta > 0$.

Lemma 3 (Boundary information). *One outcome label carries Fisher information $\mathcal{I}(d) = \beta^2/\big(4\cosh^2(\beta d/2)\big) \le \min\{\beta^2/4,\ \beta^2 e^{-\beta|d|}\}$ about θ. If all labels are successes, the log-likelihood is strictly increasing in θ and has no finite maximizer.*

Proof. With $q_\theta = \mathrm{sig}(-\beta d_\theta)$, $\partial_\theta q_\theta = \beta q_\theta(1-q_\theta)$, so $\mathcal{I} = (\partial_\theta q_\theta)^2/[q_\theta(1-q_\theta)] = \beta^2 q_\theta(1-q_\theta) = \beta^2/\big(4\cosh^2(\beta d/2)\big)$, and $(1+e^a)^2 \ge e^{2a}$ gives the tail. With successes only, $\sum_i \log \mathrm{sig}\big(\beta(\theta - d_0(z_i))\big)$ increases in θ. □

Remark 3 (Scope). Information adds over trajectories, and failures carry most of it because successes lie mostly inside the success region. Under Assumption 1(b) with $C_\sigma = C_0 + \sigma^2 I$, the posterior logit at $\sigma \to 0$ has gradient $u_0 = C_0^{-1}\Delta\mu$, so $\beta = \|u_0\|$: sharp boundaries make labels informative and, by lowering ρ, keep the contrast aligned. App. E fits β on held-out robot outcomes.

A.6 Long-Horizon Extension

This extension explains why CALM, trained with trajectory labels only, still acts on the stage where a failure begins. Split a trajectory latent into stages $x = (x^{(1)}, \ldots, x^{(S)})$ with prefix $h_k = (x^{(1)}, \ldots, x^{(k-1)})$. The stage-local contrast $g_k^{\mathrm{loc}} = s_k^+ - s_{k,k}^-$ compares, under the same prefix, the stage-k score of successes with that of failures whose onset is stage k; if Assumption 1 holds for $x^{(k)}$ given h_k, Prop. 2 applies to it with ratio $\rho_k = \delta_k/\|u_k\|$. Trajectory labels never expose onset-conditioned densities, so g_k^{loc} is only a target, and the proposition bounds how close the trajectory-level contrast comes to this stage-local target.

Proposition 4 (What the trajectory-level contrast sees in stage k). *Let $G = \nabla_x \log(p_\sigma^+/p_\sigma^-)$ under trajectory labels and $s_{\mathrm{mix},k}^-$ the stage-k failure score with unknown onset. The stage-k block of G is*

$g_k^{\mathrm{loc}} + a_k + b_k$, with latent-onset term $a_k = s^-_{k,k} - s^-_{\mathrm{mix},k}$ and suffix coupling $b_k = \nabla_{x^{(k)}} \log\big(p_\sigma(x^{>k} \mid x^{\le k}, c^+)/p_\sigma(x^{>k} \mid x^{\le k}, c^-)\big)$. If the onset posterior puts weight at least $1-\nu$ on stage k and other onsets change the stage-k score by at most B_k, then $\|a_k\| \le \nu B_k$. Let $\eta_k < \|u_k\|$ bound the task-critical part of $a_k + b_k$, $\bar{\delta}_k$ the scaffold part of the stage-k block, and $\delta_{<k}, \delta_{>k}$ the blocks of the other stages. With $\tilde{u}_k$ equal to u_k in the task-critical coordinates of stage k and zero elsewhere,

$$\cos\angle(G, \tilde{u}_k) \;\ge\; \frac{\|u_k\| - \eta_k}{\sqrt{\|u_k\|^2 + \bar{\delta}_k^2 + \delta_{<k}^2 + \delta_{>k}^2} + \eta_k}.$$

Proof. Differentiate $\log p_\sigma(x \mid c) = \log p_\sigma(x^{<k} \mid c) + \log p_\sigma(x^{(k)} \mid h_k, c) + \log p_\sigma(x^{>k} \mid x^{\le k}, c)$ in $x^{(k)}$: the prefix term vanishes, the middle terms give $g_k^{\mathrm{loc}} + a_k$, and the suffix terms give b_k. The failure density is a mixture over onsets, so $s^-_{\mathrm{mix},k}$ is the responsibility-weighted average of the $s^-_{j,k}$, and terms with $j \neq k$ carry total weight at most ν. The angle bound follows as in Lemma 2, with $a_k + b_k$ as the error and the blocks of the other stages in the norm. □

The bound makes two predictions, which App. D tests on RoboCasa composites with oracle and shuffled stage masks. (1) Stages that a failure shares with successes contribute small blocks, so the stage-agnostic contrast already points at the onset stage. (2) Masking G to the correct stage removes $\delta_{<k}$ and $\delta_{>k}$ and tightens the bound, whereas a mask on a wrong stage discards u_k.

B Last-Mile Audit Protocol

This appendix specifies the audit of Sec. 3.1 and applies it to our corpus (Tab. 7), to constructed off-scaffold executions, and to public real-robot and simulated data.

Table 7: **Last-mile audit of our real-robot failures.** Last-mile over procedure-complete failures under task-specification tolerances, and mean pre-retract coverage; App. B.4 gives REASSEMBLE.

	Tube	Peg	Card	Cup-H	Block	Cup-S	**All**
Ours: teleoperated failures							
Last-mile / procedure-complete	15/16	22/23	18/19	14/14	27/28	31/32	**127/132**
Pre-retract coverage (%)	72.4	83.1	76.6	77.5	81.1	82.9	**78.9**

B.1 Audit Algorithm

The same algorithm scores our teleoperated failures, the constructed executions, and every public corpus, per task and per tolerance rule. Positions are in centimeters, and a rotation residual is the rotation vector of $\bar{R}_k^\top R_k$ in degrees, in the scaffold's tool frame.

Audit of one task. *Input:* successes $\mathcal{S}$, procedure-complete failures $\mathcal{F}$, and optional control executions, each with poses $x_{0:T} = (p_t, R_t)_{t \le T}$ and anchors $t_1 < t_2$. Constants: K=400, W=10, threshold 0.6.

1. *Resampling.* Split each trajectory into approach $[0, t_1)$, interaction $[t_1, t_2)$, and retract $[t_2, T]$. Resample each phase to a fixed number of steps set by the median phase fractions of $\mathcal{S}$: K steps in total, K_{t_2} of them before release.
2. *Scaffold.* At each step k, $\bar{p}_k$ is the mean success position and $\bar{R}_k$ the chordal mean success rotation; the residual is $e_k(x) = \big(p_k - \bar{p}_k,\ \log(\bar{R}_k^\top R_k)^\vee\big) \in \mathbb{R}^6$.
3. *Tolerance* $\sigma_k \in \mathbb{R}^6$, from one of three rules. *Specification* (our tasks): fixed per task, phase, and degree of freedom (Tab. 8). *stepwise95* (public corpora): per-step RMS of e_k over $\mathcal{S}$, smoothed over ± 10 steps within the phase, floored at 0.05 cm and 0.25°, and scaled per phase so that 95% of success samples lie inside. *joint95* (comparison): as stepwise95 with one shape per phase.
4. *Distance and divergence.* $d_k(x) = \|e_k(x) \oslash \sigma_k\|$, so $d_k \le 1$ inside the envelope; k_{div} is the first $k \le K_{t_2} - W$ with $d_{k'} > 1$ for all $k \le k' < k + W$, and K_{t_2} if there is none.
5. *Scores.* $C_{\mathrm{int}} = k_{\mathrm{div}}/K_{t_2}$ (Eq. (1)); a procedure-complete trajectory is last-mile if $C_{\mathrm{int}} \ge 0.6$. The terminal residual is $\|p_k - \bar{p}_k\|$ at the last pre-retract step, and the pre-retract RMS distance is $d_{\mathrm{traj}} = \big(K_{t_2}^{-1} \sum_{k \le K_{t_2}} d_k^2\big)^{1/2}$ in tolerance units.
6. *Held-out successes* (success-defined rules). For each $s \in \mathcal{S}$, refit scaffold and tolerance on $\mathcal{S} \setminus \{s\}$ and score s. The conformance ratio is $\Gamma = \Pr[C_{\mathrm{int}} \ge 0.6 \mid \mathcal{F}] \,/\, \Pr[C_{\mathrm{int}} \ge 0.6 \mid \text{held-out } \mathcal{S}]$. Controls are scored against the task's scaffold and envelope, and the conservative rate also counts the screened-out failures as not last-mile.

Anchors and stages. On our tasks and on REASSEMBLE, t_1 is the onset of the final gripper closure on the object and t_2 the onset of its release. In simulation, public object predicates define the windows, and successes still define the tolerances. A multi-stage task gets anchors, a scaffold, and an envelope per stage, and the audit runs per stage.

Tolerance sources and intervals. Task specifications fix what a task tolerates, so on our tasks the last-mile rate is read directly. A success-defined envelope passes a share of successes set by how tightly they agree, and Γ removes this dependence by scoring held-out successes under the same rule. We fixed stepwise95 from the spread of successes before computing any failure statistic and apply it unchanged to every public source, including the controls. Rates carry Wilson 95% intervals; intervals for Γ come from a percentile bootstrap with 2000 replicates, which resamples recordings on REASSEMBLE, where failures cluster within a session, and trajectories within each task elsewhere.

B.2 Our Corpus

Collection and procedure screen. Experienced operators teleoperate the Franka FR3 with a leader arm. Each task follows one fixed procedure: approach, grasp, carry, insert, slot, hang, or stack, release, and retract. Every attempt receives a terminal success or failure label under success criteria fixed before collection, and this label is the only supervision that CALM uses. Four event rules remove attempts without a complete procedure; they read the logged events and never the audit metric: an *early drop* (the object leaves the gripper before the target), a *missed target*, an *interruption* of the recording before release, and a *reset* of the scene before release. Of 657 attempts (505 successes and 152 failures), the screen removes 20: 7 early drops, 5 missed targets, 4 interruptions, and 4 resets. The retained 637 trajectories hold 132 procedure-complete failures. Training draws 40S+10F per task from them by fixed trajectory lists; no audit statistic enters this selection.

Tolerances. Tab. 8 lists the specification tolerances, fixed per task and phase before any failure was inspected: loose during the approach and tight at contact, down to a few millimeters and degrees on the tightest tasks. It also lists the largest residual of the failures after divergence, the worst departure inside the task-critical window, which Fig. 1b draws normalized by the cross-task maxima.

Sensitivity. Scaling every specification tolerance by $\times 0.5$, $\times 1$, and $\times 1.5$ barely changes the last-mile count (Sec. 3.1). Mean pre-retract coverage is then 73.8, 78.9, and 82.0% for natural failures against 38.7, 41.9, and 45.0% for constructed executions: the tolerances set the absolute numbers, and natural failures keep roughly twice the coverage of constructed executions at every scale.

Table 8: **Specification tolerances and residuals of our tasks.** Tolerances per phase in (x, y, z) and (roll, pitch, yaw). Residuals: largest deviation after divergence; bold: cross-task maximum.

Task	Max. residual (cm, °)				Stage tolerance		
	Pos	Roll	Pitch	Yaw	Phase	σ_{pos} (cm)	σ_{rot} (°)
Tube	4.1	**5.7**	**7.8**	5.9	Approach	$(1.0, 1.0, 3.0)$	$(15.0, 10.0, 15.0)$
					Transit/interact	$(0.5, 0.5, 1.0)$	$(2.0, 5.0, 5.0)$
Peg	2.4	4.0	4.3	4.3	Approach	$(4.0, 4.0, 4.0)$	$(25.0, 25.0, 25.0)$
					Transit/interact	$(1.0, 1.0, 2.0)$	$(5.0, 5.0, 10.0)$
Card	2.1	3.0	3.5	**7.2**	Approach	$(5.0, 5.0, 5.0)$	$(15.0, 15.0, 15.0)$
					Transit/interact	$(0.4, 0.4, 0.5)$	$(2.0, 2.0, 1.5)$
Cup-H	2.6	3.8	7.1	5.9	Approach	$(2.0, 2.0, 2.0)$	$(20.0, 20.0, 20.0)$
					Transit/interact	$(0.5, 1.5, 1.0)$	$(5.0, 10.0, 5.0)$
Block	4.3	2.1	5.2	5.8	Approach	$(3.0, 3.0, 3.0)$	$(20.0, 25.0, 20.0)$
					Transit/interact	$(1.0, 1.0, 1.0)$	$(10.0, 10.0, 10.0)$
Cup-S	**7.8**	4.1	7.4	3.3	Approach	$(4.0, 4.0, 4.0)$	$(25.0, 25.0, 25.0)$
					Transit/interact	$(2.0, 2.0, 2.0)$	$(10.0, 10.0, 10.0)$

B.3 Constructed Off-Scaffold Executions

Constructed executions are a stress set whose membership is fixed by how they were produced, before any metric is computed; the audit never selects or removes a member. Ours are 60 executions, 10 per task, each a scripted replay of a held-out success of the same task. The replay is unchanged until the gripper closes; a fixed offset in translation, height, or rotation, of 3 to 9 times the interaction tolerance of the task, is then ramped in and held until release. All 60 executions fail under the success criterion of their task. They enter training only in the failure-source and contamination experiments (Sec. 5.4). Their pre-retract distance has an interquartile range of 2.70–5.10 tolerance units, against 0.81–1.43 for procedure-complete failures, so the two interquartile ranges do not overlap. A last-mile failure with a short terminal miss can combine high coverage with a d_{traj} slightly above one, since d_{traj} averages the whole window. Two public control sets play the same role in App. B.4: *wrong-target successes*, successful REASSEMBLE insertions into the nearest other hole scored against the scaffold and envelope of the task, and the 100 *tosses* of robomimic Can-Paired, in which the operator grasps the can and deliberately tosses it out of the workspace.

B.4 Public-Data Audit

We apply the audit, with the thresholds used on our corpus, to public corpora whose failures we did not produce: REASSEMBLE (Sliwowski et al., 2025), haptic teleoperation of insertions on a real Franka FR3, where 147 of 212 failed insertions pass against 587 of 811 held-out successes ($\Gamma = 0.96$), and, as the exploration contrast, the robomimic Can-MG rollouts of a reinforcement-learning agent at several points of its training (Mandlekar et al., 2021).

REASSEMBLE units and anchors. We read the proprioceptive streams from a public LeRobot port of the dataset at 30 frames per second, in the frame of the assembly board registered per recording; successful release points scatter by a median of 11.2 mm around their task median, which bounds operator variation and registration error together. A unit is a successful pick followed directly by the insertion of the same object, and the insertion label is its outcome; t_1 is the onset of the final gripper closure of the pick, and t_2 the onset of release, when the gripper opens by more than 5 mm. Each of the 17 tasks has enough successes to fit its own scaffold and envelope.

Screening and released failures. Of the 246 failed insertions, 34 lack a complete procedure: 28 are retries without a new grasp, 5 have no preceding pick, and 1 follows a failed pick. The same rules remove 106 successes, and conservative accounting leaves 147 of 246 failed insertions last-mile (59.8%). In 127 of the 212 procedure-complete failures the operator keeps holding the object and retries in a new segment, so t_2 falls at the annotated end of the attempt. Without these units, and with every envelope refit, 71 of 85 released failures are last-mile (83.5%) against 72.0% of held-out successes ($\Gamma = 1.16$, 95% CI [1.03, 1.29]). They end 1.14 cm from the scaffold, as close as held-out successes (1.20 cm): each released failure misses at contact, after an otherwise correct motion.

Table 9: **REASSEMBLE audit per insertion task.** Passing over scored trajectories (stepwise95); median terminal residuals of failures (F) and held-out successes (S); gap to the nearest other hole.

Insert task	n_S	n_F	stepwise95, passing / scored			Resid. F / S (cm)	Nearest hole (cm)
			Last-mile failures	Held-out successes	Wrong-target successes		
BNC	47	23	15/23	28/47	0/46	3.61 / 1.23	6.9
Bolt 4	45	23	9/23	30/45	0/48	4.93 / 1.02	10.8
D-SUB	50	8	7/8	30/50	4/39	3.10 / 1.16	8.2
Ethernet	47	18	10/18	31/47	8/50	3.94 / 1.51	10.2
Large gear	57	6	1/6	43/57	4/50	2.34 / 1.33	4.5
Medium gear	50	8	6/8	41/50	0/47	3.19 / 1.32	3.7
Round peg 1	46	10	9/10	32/46	0/47	1.42 / 1.16	6.9
Round peg 2	47	10	10/10	35/47	5/45	1.30 / 1.23	7.4
Round peg 3	45	13	8/13	34/45	8/46	1.43 / 1.35	7.6
Round peg 4	48	7	7/7	40/48	7/50	1.30 / 0.98	8.2
Small gear	47	6	5/6	41/47	1/50	2.24 / 1.14	3.7
Square peg 1	39	17	15/17	27/39	26/55	1.01 / 1.15	7.6
Square peg 2	50	11	7/11	30/50	0/47	2.26 / 1.20	7.7
Square peg 3	46	12	9/12	33/46	13/55	2.25 / 1.33	6.8
Square peg 4	45	14	13/14	34/45	10/47	1.70 / 1.36	7.4
USB	47	18	11/18	38/47	7/45	4.87 / 1.10	8.6
Waterproof	55	8	5/8	40/55	1/46	4.33 / 0.90	6.8
All	811	212	147/212	587/811	94/813	2.47 / 1.20	3.7–10.8

A per-step envelope. Successes grasp each object wherever it lies and converge on one hole: the per-task spread of the grasp location has a median of 28.9 and 26.3 cm in x and y, against 1.9 and 1.8 cm at release. The stepwise95 envelope follows this spread step by step. A phase-constant envelope (joint95) carries the grasp spread into the interaction phase, where its semi-axes of 43 to 74 cm far exceed the 3.7 to 10.8 cm between neighboring holes; it gives a similar ratio for failed insertions ($\Gamma = 0.97$) but also passes 33.9% of wrong-target successes.

Wrong-target control and per-task results. Wrong-target successes (App. B.3) have a median C_{int} of 0.19, against 1.00 for failed insertions, so they leave the envelope early in the approach, and they end 7.80 cm from the scaffold. Tab. 9 lists every insertion task; per-task failure counts are small, so the finding rests on the pooled rate, whose recording-level bootstrap over 83 recordings bounds the failure-minus-success difference to $[-8.8, +2.3]$ points. Failed peg insertions pass at least as often as held-out successes in seven of the eight peg tasks, whereas connector and bolt insertions account for the largest failure residuals of the corpus.

Exploration failures conform less than successes. Can-MG holds 3900 rollouts of a SAC agent saved at 13 checkpoints, grouped in training order into blocks of 300. The agent's gripper commands switch a median of 26 times per rollout, so object predicates set the anchors: t_1 when the can is lifted at least 1 cm and held within 5 cm of the end effector, and t_2 when it is next more than 6 cm away. Of the 3182 failures, 422 are procedure-complete (the others never lift the can, hold it until the horizon, or have degenerate anchors), and held-out successes are a fixed random 300 of the 705 procedure-complete successes. These failures are last-mile at 35.5% against 80.0% for held-out successes ($\Gamma = 0.44$, 95% CI [0.39, 0.51]) and end 45.08 cm from the scaffold, against 9.71 cm. Over training, successes rise from none in the earliest 6 blocks to 67.7% in the last, and the median C_{int} of failures rises with them, yet the last-mile share of every block stays below the pooled held-out success rate: as the agent improves, its failures move toward the scaffold.

Can-Paired tosses and sensitivity. Coverage records *when* a trajectory leaves the envelope, and the terminal residual *where* it ends. On Can-Paired the approach fills 54.2% of the pre-retract window, so 28 of the 100 tosses still reach $C_{\text{int}} \geq 0.6$, but none ends within the 95% quantile of the terminal residuals of held-out successes (62.38 against 4.22 cm). Scaling every tolerance by $\times 0.5$ to $\times 1.5$ and setting W between 5 and 20 steps moves the absolute rates, since the envelope sits at a fixed quantile of the success spread, but in every setting and under both rules REASSEMBLE failures conform more than Can-MG failures and more than wrong-target successes.

C Implementation Details

This appendix gives the configuration of CALM, the implementation of every baseline, control, and variant, and the protocol of the latent diagnostics. Tab. 10 lists the default configuration; the Cosmos-based baselines and controls share its backbone and optimizer.

Table 10: **CALM configuration.** Settings of the default real-robot model; backbone and recipe follow Agarwal et al. (2025) and Kim et al. (2026).

Setting	Value	Setting	Value
Model and data		**Optimization**	
Backbone	Cosmos-Predict2-2B, Cosmos Policy recipe	Objective	EDM loss, Eq. (5) only
Outcome tokens	c^+, c^-, `[NULL]`, all learned	Iterations	40K
Outcome dropout	p_{drop}=0.15, to `[NULL]`	Learning rate	1×10^{-4}, 1000 warm-up steps
Data per task	40S+10F, screened pool	Schedule	cosine over 30K steps, then flat
Mini-batch	successes : failures 2:1	Effective batch size	192
Target routing	70/30, action chunk / future frame	Hardware	4 NVIDIA H200
Inference			
Sampler	EDM, 5 steps, action horizon H=10	Guidance	$(2, 1)$, two branches
Control	executes two of H waypoints at 10 Hz; replans at 5 Hz	Hardware	RTX 4090; timing also on RTX 5090

C.1 Baselines, Controls, and Variants

Success-only and imitation. Every method draws its 50S or 40S+10F from the same per-task corpus, with failures from the procedure-screened pool (App. B). Cosmos Policy (Kim et al., 2026) is the configuration of Tab. 10 without outcome tokens, and $\pi_{0.5}$ (Black et al., 2025a) is fine-tuned on the same 50S. Cosmos Policy on 40S trains on the successes of the 40S+10F set and isolates the success count. The mixed-imitation rows train both policies on the pooled 40S+10F and imitate every failure as a demonstration, with no outcome label.

Preference. Best-of-N samples N=5 candidate action chunks per call from Cosmos Policy on 40S+10F and executes the one scored highest by an outcome predictor learned from the same trajectories. DPO starts from $\pi_{0.5}$ fine-tuned on the 40 successes of the same 40S+10F set, which also serves as the frozen reference policy, and optimizes it on success–failure pairs of the same task, with the successful trajectory preferred (Rafailov et al., 2023; Zhang et al., 2024).

Controls without failure data. Degenerate outcome-CFG trains on 50S with every trajectory labeled c^+ and the outcome dropout of OCT, and samples outcome guidance toward c^+. Both branches then model the same data, so $q_\sigma \equiv 1$ and $s_+ - s_\varnothing = \nabla_x \log q_\sigma = 0$: the row measures what an extra outcome branch adds by itself. Context-CFG drops the task context to a null context during training on 50S and samples classifier-free guidance on it (Ho & Salimans, 2022), which sharpens the success distribution without any failure.

Outcome-classifier control. A noise-conditional classifier $q_\phi(c^+ \mid x_\sigma, \sigma, \text{ctx})$ on the Cosmos backbone is trained with binary cross-entropy on noised latents of the same 40S+10F. It guides a Cosmos Policy denoiser $\epsilon_\varnothing$ of the pooled 40S+10F with the potential of CALM at $(2, 1)$, $\hat{\epsilon} = \epsilon_\varnothing - \sigma\nabla_x[\lambda_+ \log q_\phi - \lambda_- \log(1 - q_\phi)] = \epsilon_\varnothing - \sigma\kappa(x)\nabla_x\hat{\ell}_\phi$, with classifier logit $\hat{\ell}_\phi$ and the gain κ of Prop. 1 at q_ϕ. At the population optimum this is the rule of CALM (App. A.1); in practice the two differ in how they estimate it, a classifier on top of a pooled denoiser versus the two conditional branches of a single outcome-conditioned denoiser.

Guidance rules and contrast forms. Every rule in Tabs. 15 and 17 reuses the CALM checkpoints and changes only (λ_+, λ_-) in Eq. (6). The action-only variant of Tab. 19 routes every training sample to the action target and keeps all other settings of Tab. 10. Four variants change how the contrast enters. (1) *Without* `[NULL]`: OCT trains on c^+ and c^- only and samples $2\epsilon_+ - \epsilon_-$, which needs no null branch. (2) *Hinge* rectifies the contrast element-wise, $\hat{\epsilon} = \epsilon_\varnothing + \lambda \max(0, \epsilon_+ - \epsilon_-)$.

(3) *KL-regularized* and (4) *reverse-KL* training add $D_{\mathrm{KL}}(p_+ \,\|\, p_-)$ or $D_{\mathrm{KL}}(p_- \,\|\, p_+)$ to the OCT loss, approximated at the current noisy sample in each training step of OCT.

C.2 Latent Diagnostics

Field and reference. Every diagnostic reads the learned field of the model it describes, $\hat{g} = (\epsilon_- - \epsilon_+)/\sigma$, an estimate of $g_\sigma = s_+ - s_-$. A fixed linearization J_E of the encoder maps the signed physical residual between the success and failure means in the task-critical window to latent tokens, $\Delta\mu_z = J_E r_{\mathrm{phys}}$. An uncontaminated calibration split, held out from training, fixes $C_\sigma = \Sigma_z + \sigma^2 I$ from the clean residual covariance Σ_z of the task-critical tokens, $u = C_\sigma^{-1}\Delta\mu_z$, and the scaffold and task-critical token masks shared by all diagnostics.

Queries and metrics. Queries $(x_\sigma, \sigma, \mathrm{ctx})$ come from trajectories disjoint from the calibration split, with fixed noise draws, at a fixed low-to-mid noise level unless a table names the band. Each query gives the scaffold ratio $\hat{\rho} = \|\hat{g}^{\mathrm{sc}}\|/\|u\|$ and the angle between $\hat{g}$ and $\bar{u} = (0, u)$. Per-task diagnostics use the native Euclidean metric on latent tokens, in which the scaffold and task-critical blocks are orthogonal. The contamination study uses the covariance metric $\|v\|^2_{C_\sigma} = v^\top C_\sigma v$ with its matching cosine, in which $\|u\|^2 = \Delta\mu_z^\top C_\sigma^{-1}\Delta\mu_z$ is the squared Mahalanobis separation of the outcome means, and reports the mean over the six real-robot tasks; both metrics give $\hat{\rho} < 1$ on the natural pool. Across contamination levels, u, C_σ, the masks, the queries, the noise draws, and the evaluation initializations stay fixed, so each row measures its own learned field against one reference.

D Simulation Benchmarks

This appendix tests whether the gain carries over to public simulation suites, whose contact is cleaner and whose success rates are higher, under matched data, recipe, and compute.

Table 11: **LIBERO-Spatial.** Success (%). Controlled rows train on Spatial only with matched recipe, compute, and episodes; the published row is not controlled.

Method	Data	Succ. (%)
Controlled *Spatial-only training*		
Cosmos Policy	450S	95.4
Cosmos Policy	500S	96.0
Mixed imitation	450S+50F	92.8
Best-of-N	450S+50F	96.6
CALM	450S+50F	97.8
CALM	500S+50F	**98.2**
Published *not controlled*		
Cosmos Policy	all four suites	98.1
Low budget *100 demonstrations*		
Cosmos Policy	demos only	76.2
CALM	+ policy failures	**86.8**

Table 12: **RoboCasa-24.** Success (%) over the 24 atomic tasks, 50 demonstrations per task.

Method	Data	Succ. (%)
Cosmos Policy, ours	50S	66.9
CALM	40S+10F	**68.3**
Cosmos Policy, published	50S	67.1

Table 13: **RoboCasa composites.** Success (%) by stage count, 240 episodes per group; oracle and shuffled masks.

Stages	Cosmos	CALM	Oracle	Shuffled
2–3	61.3	67.1	69.6	60.8
4–5	43.3	51.7	55.4	42.9
≥6	26.7	35.0	39.2	25.8
All	43.8	51.3	54.7	43.2

LIBERO-Spatial. On LIBERO-Spatial (Liu et al., 2023a) with equal demonstrations (Tab. 11), CALM on 450S+50F reaches 97.8%, 1.8 points above Cosmos Policy on 500S, and adding the same 50 failures to 500S gives 98.2% (+2.2). Mixed imitation falls to 92.8% and Best-of-N reaches 96.6%, the same ordering as on the real robot. The published Cosmos Policy reaches 98.1% with co-training on all four suites and more data and compute (Kim et al., 2026). The gain is larger where success is lower: at the low budget, the policy's own failures raise success from 76.2% to 86.8%.

RoboCasa-24. On RoboCasa (Nasiriany et al., 2024), CALM trains on 40 successes and 10 policy failures per task. It gains 1.4 points over our reproduction of Cosmos Policy and improves success on 18 of the 24 tasks (Tab. 12); the published Cosmos Policy reports 67.1% (Kim et al., 2026).

RoboCasa composites. Tab. 13 groups twelve composite RoboCasa tasks by their number of stages. Default CALM, with one label per trajectory, improves over Cosmos Policy in every group, from

Table 14: **Predictions and their tests.** Rows P3 to P7 test the four predictions of Sec. 3.3; P1 tests the audit on public data and P2 the label information, and all rows but P1 use our real-robot data. ✓: the measured result shows the predicted direction of change.

#	Prediction (source)	Test	Result	Observed
P1	Natural failures conform like successes (Sec. 3)	Γ: REASSEMBLE vs. robomimic	0.96 vs. 0.44	✓
P2	Label information peaks at the boundary (Lem. 3)	Held-out logistic fit	β=2.3±0.4; NLL 0.31 vs. 0.51	✓
P3	Scaffold contrast stays below task-critical separation (Prop. 2)	Native scaffold ratio, six tasks	$\hat{\rho} \leq 0.52$	✓
P4	Off-scaffold failures break alignment and success (Rem. 2)	Replace 0 to 50% of failures	$\hat{\rho}$ 0.41→1.08; 77.2→62.8%	✓
P5	Both gains matter (Prop. 1)	$(2,0)$, $(0,1)$ vs. $(2,1)$	Δ −13.9, −20.0	✓
P6	Guidance is inert without failures (Prop. 1)	Degenerate outcome-CFG vs. Cosmos	Δ +0.5	✓
P7	Trajectory labels suffice (Prop. 2)	Trajectory vs. step-wise labels	89.7% of the gain	✓

43.8% to 51.3% overall and from 26.7% to 35.0% with six or more stages. Oracle stage masks add a smaller further gain (54.7%), and masks on shuffled stages fall back to the level of Cosmos Policy (43.2%). Both results match the two predictions of App. A.6: the stage-agnostic contrast gives most of the gain, and the extra gain needs the correct stage.

Simulated policy failures are last-mile. With tolerances set by LIBERO successes and windows set by object predicates (App. B), a frozen low-budget Cosmos Policy checkpoint produces 100 procedure-complete failures on initializations disjoint from evaluation, of which 82 are last-mile, with median $d_{\text{traj}} = 1.16$ and 73.6% coverage; scripted off-scaffold executions give 3.21 and 31.4%, and 8 of 100 pass. On RoboCasa-24, 157 of the 240 policy failures used for training are last-mile, so in both suites most policy failures lie in the regime where CALM improves over matched controls.

E Additional Real-Robot Results

This appendix describes the real-robot tasks and gives the full results behind Sec. 5, together with the latent diagnostics, the inference cost, and the data budget.

E.1 Tasks and Rollouts

Tasks. All experiments run on the 7-DoF Franka Research 3 arm of Fig. 3, observed by a wrist-mounted RealSense D405 and a front-view RealSense D435. Four experienced operators teleoperate the arm with a leader–follower interface, follow a fixed procedure per task, and give every attempt a terminal label defined before collection. Fig. 4 shows the six tasks. A rollout succeeds when the object ends in the instructed configuration after release, and each method is evaluated with 30 rollouts per task on matched initial configurations. The last-mile failures of each category share one mode: a small pose error blocks the insertion, a small orientation or contact mismatch prevents the placement in the slot or rack, or a small release or support deviation leaves the completed stack unstable.

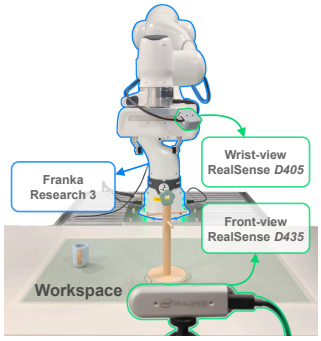
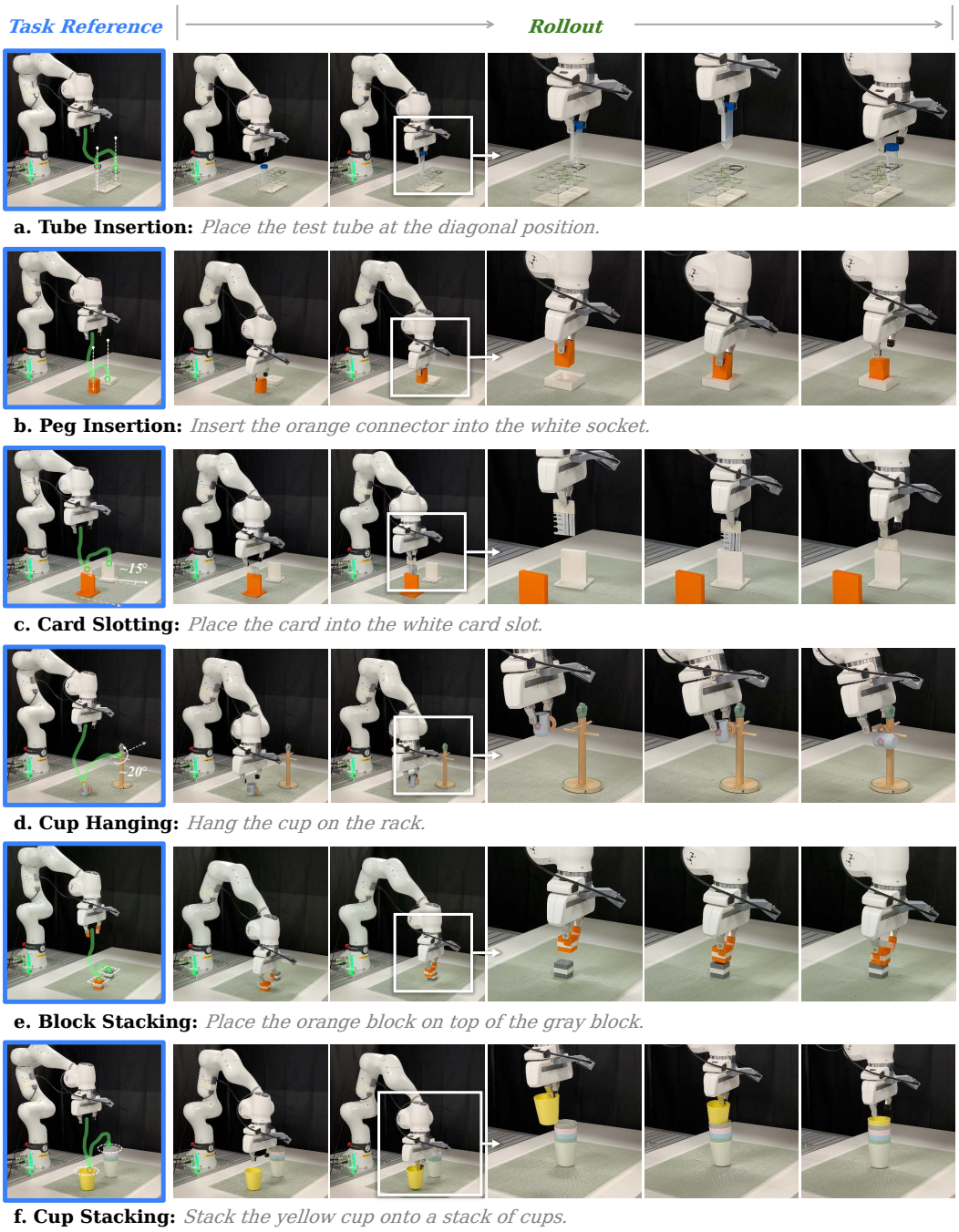


Figure 3: **Setup.** Franka Research 3 with wrist D405 and front D435 cameras.

E.2 Guidance Rules, Training Forms, and Data Budget

All rules lie on one family. The two-branch rule $(1+w)\epsilon_+ - w\epsilon_-$ of prior work (Ban et al., 2024; Liang et al., 2026; Zheng et al., 2026) is the null-free line $\lambda_- =$



a. Tube Insertion: *Place the test tube at the diagonal position.*

b. Peg Insertion: *Insert the orange connector into the white socket.*

c. Card Slotting: *Place the card into the white card slot.*

d. Cup Hanging: *Hang the cup on the rack.*

e. Block Stacking: *Place the orange block on top of the gray block.*

f. Cup Stacking: *Stack the yellow cup onto a stack of cups.*

Figure 4: **Real-robot tasks.** Goal configuration, CALM rollout frames, and instruction per task.

w: its w=1 case is the default $2\epsilon_+ - \epsilon_-$, so the two share one measurement (139/180), and its w=2 case is $(3, 2)$. Attraction-only is the rule of optimality-token guidance (Frans et al., 2025; Physical Intelligence et al., 2025), repulsion-only that of negative guidance (Ban et al., 2024), and the balanced rule the negation of composable diffusion (Liu et al., 2022); evaluating all of them on one checkpoint compares the rules themselves, not the training pipelines of those methods. The balanced rule $(1, 1)$ reaches 71.7%, strong attraction $(3, 1)$ 76.1%, and $(3, 2)$ 72.2%; all three stay well above every single-gain rule, so combining both gains matters more than the exact value of either weight. The success branch alone, $(1, 0)$, samples the default checkpoint without guidance

Table 15: **Guidance rules as points of one family.** Every rule samples the same CALM checkpoints with $\hat{\epsilon} = (1-\lambda_+ + \lambda_-)\epsilon_\varnothing + \lambda_+\epsilon_+ - \lambda_-\epsilon_-$; 180 rollouts per rule. Δ: default minus row.

Rule	(λ_+, λ_-)	Null coef.	Prediction $\hat{\epsilon}$	Succ. (%)	Δ
Null branch only	$(0, 0)$	1	$\epsilon_\varnothing$	47.2	+30.0
Success branch only	$(1, 0)$	0	ϵ_+	59.4	+17.8
Attraction-only	$(2, 0)$	-1	$2\epsilon_+ - \epsilon_\varnothing$	63.3	+13.9
Repulsion-only	$(0, 1)$	2	$2\epsilon_\varnothing - \epsilon_-$	57.2	+20.0
Balanced	$(1, 1)$	1	$\epsilon_\varnothing + \epsilon_+ - \epsilon_-$	71.7	+5.5
Default (two-branch, w=1)	$(2, 1)$	0	$2\epsilon_+ - \epsilon_-$	**77.2**	*ref.*
Strong attraction	$(3, 1)$	-1	$3\epsilon_+ - \epsilon_\varnothing - \epsilon_-$	76.1	+1.1
Two-branch, w=2	$(3, 2)$	0	$3\epsilon_+ - 2\epsilon_-$	72.2	+5.0

Table 16: **Full two-network reproduction.** Every row trains on the same 40S+10F per task and is evaluated on the same 180 initializations. Success (%) with successes/rollouts.

Method	Trained networks	Succ. (%)
$\pi_{0.5}$ + DPO	one policy	64.4 (116/180)
DIPOLE-style reproduction (Liang et al., 2026)	two policies and a critic	66.1 (119/180)
Outcome classifier	one policy and a classifier	68.3 (123/180)
CALM	one policy	**77.2** (139/180)

Table 17: **Guidance sweep.** One weight varies around the default $(2, 1)$ on the same checkpoints.

λ_+	λ_-	Null coef.	Successes	Succ. (%)
λ_+ sweep at λ_-=1				
1	1	1	129/180	71.7
1.5	1	0.5	135/180	75.0
2	1	0	139/180	77.2
2.5	1	-0.5	138/180	76.7
3	1	-1	137/180	76.1
4	1	-2	136/180	75.6
6	1	-4	129/180	71.7
λ_- sweep at λ_+=2				
2	0.5	-0.5	134/180	74.4
2	1	0	139/180	77.2
2	1.5	0.5	137/180	76.1
2	2	1	137/180	76.1
2	2.5	1.5	132/180	73.3
2	3	2	128/180	71.1
2	4	3	112/180	62.2

Table 18: **Training and contrast form (top); data budget (bottom).**

Variant	Successes	Succ. (%)
`[NULL]` in training		
With `[NULL]` (default)	139/180	**77.2**
Without `[NULL]`	133/180	73.9
Contrast form		
Linear (default)	139/180	**77.2**
KL-regularized	121/180	67.2
Hinge	119/180	66.1
Reverse KL	117/180	65.0

Method	N=30	N=50
Baselines *two-backbone means*		
Success-only	44.4	56.4
Mixed imitation	36.7	46.4
Trajectory preference	50.0	61.1
CALM	**66.7**	**77.2**

and reaches 59.4%, against 48.9% for Cosmos Policy on the same 40 successes; attraction-only guidance then adds 3.9 points and the default a further 13.9.

A full two-network reproduction. Tab. 15 evaluates the two-branch rule of DIPOLE (Liang et al., 2026) on the CALM checkpoint; Tab. 16 also reproduces its training on the robot. Two Cosmos Policy networks are trained for 40K steps each with the loss of the official code, the critic uses the backbone of the outcome classifier with the terminal outcome as return, β keeps the DIPOLE default, and $\omega \in \{0.5, 1, 2\}$ is selected on five held-out initializations per task that are never used for evaluation. The reproduction reaches 66.1%, against 68.3% for the outcome classifier and 77.2% for CALM; it trains two policy networks and a critic, about twice the compute of CALM.

Attraction limits itself; repulsion must stay moderate. In Tab. 17, even $\lambda_+ = 6$ keeps 71.7%: a sample pushed across the boundary leaves the reach of λ_+. In contrast, $\lambda_- = 4$ falls to 62.2%, because the success residual shifts by λ_- displacements and moves past the success mode (App. A.2).

The null token is a training device; the linear contrast works best. Both `[NULL]` rows of Tab. 18 sample $2\epsilon_+ - \epsilon_-$, whose null coefficient is zero, so they differ only in the classifier-free-guidance dropout of training (Ho & Salimans, 2022), and their gap of 3.3 points is within evaluation noise.

Table 19: **Future-observation target.** Action-only routing (100/0) trains only the action target; all other settings as in Tab. 10. Success (%) with successes/rollouts; Δ: action only minus both targets.

Model and rule	Action + observation (70/30)	Action only (100/0)	Δ
CALM, default $(2, 1)$	**77.2** (139/180)	68.3 (123/180)	-8.9
CALM, success branch $(1, 0)$	59.4 (107/180)	56.1 (101/180)	-3.3
Cosmos Policy (50S)	52.8 (95/180)	50.0 (90/180)	-2.8

Table 20: **Contamination, full.** $\hat{\rho}$ and cos in the covariance metric, averaged over tasks; d_{traj} and within-tolerance share over the 180 executions of each row; + gate: CALM on the gated pool.

Replaced	$\hat{\rho}$	cos	Median d_{traj}	Within tol. (%)	CALM (%)	+ gate (%)
0/60 (0%)	0.41	0.90	0.90	68	77.2	77.2
6/60 (10%)	0.52	0.86	1.03	63	74.4	76.7
15/60 (25%)	0.71	0.79	1.21	54	68.9	75.6
30/60 (50%)	1.08	0.68	1.47	45	62.8	73.3

Success-only Cosmos: median d_{traj} 1.80, 36% of steps within tolerance.
The optional gate keeps 96% of natural failures and rejects 93% of constructed executions.

Hinge clips the contrast element-wise, and the two KL forms turn it into a scalar training penalty (App. C.1); all three replace the linear contrast that Prop. 1 places along one direction, which Prop. 2 ties to the task-critical correction at contact.

Fewer trajectories suffice with CALM. The bottom block of Tab. 18 trains every method with N trajectories per task and reports each baseline family as the mean of its two backbones. With N=30, CALM reaches 66.7%, above every baseline family trained with N=50 (at most 61.1%), and it leads every family at both budgets. Small budgets cover the outcome boundary sparsely, and last-mile failures add labels where label information peaks (App. E.4).

The future-observation target. With every training sample routed to the action target, CALM reaches 68.3% against 77.2% with both targets (Tab. 19), and its success branch $(1, 0)$ 56.1% against 59.4%; success-only Cosmos Policy reaches 50.0% against 52.8%. Action-only CALM reaches 68.3% against 64.4% for the strongest standard baseline, $\pi_{0.5}$ + DPO, and 50.0% for action-only Cosmos Policy. Each pair shares data, backbone, training budget, and evaluation initializations, so each difference isolates the future-observation target of the world action model.

E.3 Contamination and the Audit Gate

Contaminated models move toward success-only behavior. On the robot (Tab. 20), the median distance of executed trajectories to the scaffold grows from 0.90 to 1.47, and the share of pre-retract steps within tolerance falls from 68% to 45%. Both approach the values of success-only Cosmos Policy (1.80 and 36%), so contamination reduces the correction that clean failures provide. On the clean pool, 82% of the contrast energy of the learned field lies on task-critical blocks and 9% on scaffold blocks, with the token masks of App. C.2. The optional gate keeps only failures that pass the last-mile test of Sec. 3.1; it reads training trajectories and labels only, before training, and the last column of Tab. 20 trains CALM on the gated pool.

E.4 Latent Diagnostics

Alignment holds on every task. On the learned field of each default model, the native scaffold ratio stays between 0.39 and 0.52, and the learned contrast lies within 26.8° to 35.1° of the task-critical residual (Tab. 21). By Remark 1, each measured angle combines the scaffold ratio with the deviation $\hat{e}$ of the learned task-critical block from the reference direction, which includes the estimation error of the learned field and the error of the fitted reference; this explains why it exceeds the exact-field value $\arctan \hat{\rho}$ by several degrees on every task, and all angles stay below 45°. The outcome separation of 0.76 to 0.91 local SDs keeps the residual means apart, and the success/failure covariance gap stays at or below 0.22, as the local model of Prop. 2 requires. Card Slotting, the hardest task for every method in Tab. 1, has the largest ratio and the largest angle.

Table 21: **Latent diagnostics on the real-robot tasks.** Default model per task, native coordinates, low-to-mid noise.

Task	Cov. gap	Sep. (SD)	$\hat{\rho}$	Angle (°)
Tube	0.18	0.84	0.43	30.7
Peg	0.15	0.91	0.39	26.8
Card	0.22	0.76	0.52	35.1
Cup-H	0.20	0.79	0.49	33.4
Block	0.17	0.88	0.45	29.3
Cup-S	0.19	0.82	0.47	32.0

Table 22: **Score fit and outcome boundary.** Local reference fitted on the calibration split; five-fold held-out logistic fit on d_b.

Fitted local reference Noise band	Low	Mid	High
Relative score error	0.18	0.24	0.43
Score/residual angle (°)	24	27	41
Outcome boundary Distance bin	Near	Mid	Far
Median $\lvert d_b \rvert$	0.10	0.86	1.55
Outcome variance (norm.)	1.00	0.46	0.14
$\beta = 2.3 \pm 0.4$; NLL 0.31 vs. 0.51 (intercept only)			

Table 23: **Inference cost per action call.** One open-loop action call runs 5 EDM steps; mean $\pm$ SD. The default needs only the success and failure branches, so the three-branch time is an upper bound.

Per-call time *BF16, H=10, one action call of 5 EDM steps; 100 calls after 10 warm-ups*		
GPU	Cosmos, one branch (ms)	CALM, three branches batched (ms)
RTX 4090	58 ± 11	78 ± 14
RTX 5090	47 ± 9	63 ± 12
Frame to command (RTX 4090): 102 ± 15 ms, including capture 17 ± 3 ms and decoding 7 ± 2 ms.		

The local model fits late steps. The local reference, fitted on the calibration split and distinct from the learned field, agrees with the learned field on held-out queries with relative error 0.18 and 0.24 at low and mid noise and 0.43 at high noise (Tab. 22). Its angle to the residual grows accordingly, from 24° to 41°, so the local analysis describes the low-noise steps that set the final action.

Label information peaks at the boundary. The held-out boundary fit gives a slope $\beta = 2.3 \pm 0.4$ and a negative log-likelihood of 0.31, against 0.51 for the intercept-only model, and the outcome variance is largest in the bin nearest the boundary (Tab. 22), where last-mile failures sit (App. A.5).

E.5 Inference Cost

CALM runs in real time on consumer GPUs. CALM batches its branches into one forward pass at each of the 5 EDM steps (Tab. 23), so three branches add about a third to the single-branch cost on both GPUs, within 14.2 GB of memory. For H from 5 to 20, horizon-matched checkpoints take 77 to 79 ms per call and succeed in 66 to 69 of 90 matched rollouts.

F Extended Related Work

Guidance theory. Guidance rests on conditional-score identities (Song et al., 2021; Dhariwal & Nichol, 2021) and extends to products and ratios of scores (Du et al., 2023a), and a negative condition that overlaps the positive one also removes their shared content (Armandpour et al., 2023). Theory describes what guidance samples (Chidambaram et al., 2024; Wu et al., 2024b; Li et al., 2025b; Bradley & Nakkiran, 2024; Lehman Pavasovic et al., 2026). Our analysis adds a condition specific to outcome labels: the scaffold ratio states when the contrast, which does not depend on the class prior (Saerens et al., 2002) and whose task-critical block is Fisher's discriminant (Fisher, 1936), points at the task-critical correction of a failed trial.

Imperfect data, reinforcement learning, and preferences. Imperfect demonstrations are weighted, ranked, or filtered (Wu et al., 2019; Zhang et al., 2021; Brown et al., 2019; Hejna et al., 2025; Agia et al., 2025; Chen et al., 2025a), negative demonstrations mark what to avoid (Grollman & Billard, 2011; Kalinowska et al., 2021), and interventions supply corrective labels (Ross et al., 2011; Kelly et al., 2019; Mandlekar et al., 2020; Luo et al., 2024). Reinforcement learning fine-tunes diffusion and flow policies online (Ren et al., 2025; Zhang et al., 2025; McAllister et al., 2026; Luo et al., 2025) or guides them offline with learned values, advantages, or preferences (Peng et al.,

2019; Wang et al., 2023; Hansen-Estruch et al., 2023; Lu et al., 2023; Zhao et al., 2026a; Wallace et al., 2024; Hejna et al., 2024; Chen et al., 2025b). A single outcome advantage also penalizes the correct early actions of a failed rollout (Zhou et al., 2026); the last-mile contrast leaves them intact.

Failure data. Vision-language models and runtime monitors detect and explain failures (Liu et al., 2023b; Duan et al., 2025; Agia et al., 2024), yet barely separate success from failure at millimeter scale (Sudhakar et al., 2026; Navasardyan et al., 2026). Public corpora release labeled failures (Sliwowski et al., 2025; Khazatsky et al., 2024; Ofori-Ampofo et al., 2026; Mandlekar et al., 2021); failures made by perturbing successes (Duan et al., 2025) are not evidence for our audit.