BABYSTEPS: LEARNING TASK MEANING ALONG THE WAY

Anonymous authors
Paper under double-blind review

ABSTRACT

A robot can change its behavior after failure while retaining a mistaken interpretation of the demonstrated task. We study this *initial-belief anchoring* and introduce BABYSTEPS, which trains a vision-language supervisor to revise task interpretations from demonstrations and execution history while guiding a frozen vision-language-action executor. During offline training, a teacher uses the canonical task instruction to write a reflection on the demonstration, initial instruction, and failed rollout. We pair this reflection with the canonical instruction for supervised fine-tuning on student-visible histories, without requiring successful correction rollouts. At deployment, the supervisor generates reflections and revised instructions without teacher access. Our Qwen3.5-9B supervisor achieves 77.5% Success@5 on LIBERO tasks held out from supervisor training, exceeding Zero-shot and Instruction-only SFT by 30 and 10 percentage points, respectively. Under controlled incorrect-instruction interventions, both BABYSTEPS and a neutral-reflection control eliminate repairs that express only the implanted task, but BABYSTEPS more often recovers the true task without implanted components: 20/60 repairs versus 8/60. These results distinguish abandoning an incorrect interpretation from revising toward the demonstrated task.

1 INTRODUCTION

Learning from demonstrations (Argall et al., 2009) offers a natural way to specify robotic tasks, but standard imitation learning relies on costly action-labeled data, with instruction-conditioned systems additionally using language annotations. To scale robot learning, recent approaches increasingly turn to unlabeled "in-the-wild" observations, such as internet videos or unannotated play data. However, harnessing this data requires the robot to independently infer the human's intended goals, manipulated objects, and spatial relations purely from visual context. While recent methods use vision-language models (VLMs) to translate these unlabeled demonstrations into task specifications and executable programs (Wang et al., 2023; 2025), these semantic interpretations remain brittle. A model might misidentify a target receptacle or hallucinate a spatial constraint from an ambiguous video. When physical execution fails due to these semantic misunderstandings, the robot must rely on *self-evolving task understanding*, i.e., using its own execution experience to progressively refine its interpretation of the demonstrated intent.

Recent work uses demonstrations, interaction history, and reflection to improve robotic and embodied-agent behavior (Liu et al., 2023b; Luo et al., 2025; Duan et al., 2025; Jiang et al., 2026; Ju et al., 2026). These advances motivate a complementary question: how can a robot learn to revise an incorrect task interpretation inferred from a visual demonstration after execution failure? A supervisor may appropriately tweak its action trajectory across retries but remain anchored to an incorrect goal or object despite conflicting visual evidence, a failure mode we formalize as *initial-belief anchoring*. For example, a supervisor that misidentifies the target bowl spends every retry refining placement rather than questioning object choice. Escaping this trap requires reassessing task understanding, not just physical execution.

A correct task interpretation can still lead to execution failure because of imperfect low-level control, so execution success alone gives an incomplete signal of semantic revision. BABYSTEPS trains a vision-language supervisor to reconsider task interpretations using demonstrations and execution evidence while guiding a frozen vision-language-action (VLA) executor. During offline training, a

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.

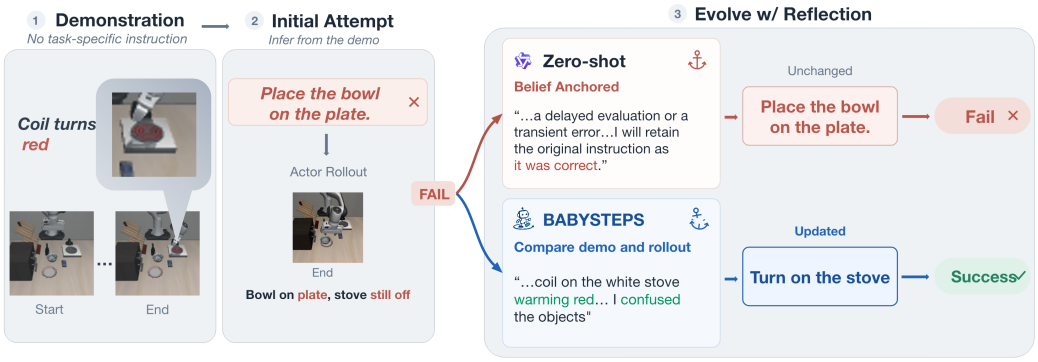

Figure 1: **Overview of the BABYSTEPS setting.** In this example, the demonstration shows a stove coil turning red, but the initial instruction asks the robot to place a bowl on a plate. After failure, the Zero-shot supervisor repeats this instruction, whereas BABYSTEPS revises it to turn on the stove and succeeds.

teacher examines the demonstration, initial instruction, and failed rollout with access to the canonical task instruction, directly authoring a reflection on the initial interpretation. We pair this reflection with the canonical instruction for supervised fine-tuning on the student's observable history; constructing these targets does not require successful correction rollouts. At deployment, the supervisor revisits the demonstration and execution history without teacher access.

Our Qwen3.5-9B supervisor achieves 77.5% Success@5 on LIBERO tasks held out from supervisor training, outperforming Zero-shot and Instruction-only SFT by 30 and 10 percentage points, respectively. A 4B supervisor trained using the same 9B-generated failure histories also reaches 77.5% Success@5. Controlled interventions with the 9B model show that BABYSTEPS and Neutral-reflection SFT, which uses generic reflection targets, both eliminate repairs that express only the implanted incorrect task. However, BABYSTEPS more often revises toward the demonstrated task, producing the true task without implanted components in 20/60 repairs versus 8/60 for the neutral control. The observed advantage is therefore in the direction of revision, beyond merely changing the initial instruction.

2 Related Work

Foundation models for robot control. Foundation models connect visual and linguistic knowledge with robotic decision making through direct action prediction and modular reasoning. RT-2 (Zitkovich et al., 2023), OpenVLA (Kim et al., 2024), and $\pi_{0.5}$ (Physical Intelligence et al., 2025b) combine vision-language pretraining with robot data to support generalizable instruction-conditioned control. Complementary approaches use foundation models to organize physical behavior: SayCan (Ahn et al., 2022) grounds skill selection in learned affordances, VoxPoser (Huang et al., 2023) constructs spatial value maps for motion planning, and Hi Robot (Shi et al., 2025) uses hierarchical vision-language reasoning to interpret complex instructions and situated feedback before low-level execution. These approaches provide a foundation for connecting task reasoning to robot actions. BABYSTEPS builds upon this foundation, pairing a VLM supervisor with a frozen VLA executor to study how demonstrated task interpretations should be revised when subsequent execution evidence challenges them.

In-context learning and reflection. In-context robot learning uses examples and interaction history to condition behavior at deployment. ICRT (Fu et al., 2024) predicts actions from sensorimotor demonstration prompts without task-specific parameter updates. Related work expands the capacity to retain context: RoboTTT (Jiang et al., 2026) compresses long visuomotor histories into recurrent fast weights updated during inference. More directly related to our setting, GPT-Policy (Cheng et al., 2026) combines visual demonstrations, goal references, and execution feedback in the context of a fixed general-purpose VLM, which selects robot-tool actions for a constrained execution

layer. Reflection provides another way to use experience: Reflexion (Shinn et al., 2023) stores verbal feedback in episodic memory to improve subsequent trials, while REFLECT (Liu et al., 2023b) summarizes robot experiences hierarchically to explain failures and support corrective planning. These works establish the value of context and feedback for adaptation and recovery. We investigate a specific failure within this setting, *initial-belief anchoring*, in which an incorrect task interpretation persists across retries despite contrary evidence. BABYSTEPS trains the supervisor to revisit inferred goals and subgoals using the demonstration and failed executions. Its teacher prompt asks for reflections that preserve interpretations supported by the evidence and acknowledge insufficient evidence; these are supervision requirements, and we do not independently evaluate retention of correct interpretations or uncertainty judgments.

Adaptation of VLA-based systems. Adaptation can improve the action policy itself or learn an upstream component that conditions its behavior. OpenVLA supports efficient supervised adaptation (Kim et al., 2024), while RL4VLA (Liu et al., 2025) studies the generalization benefits of reinforcement learning relative to supervised fine-tuning. RECAP, used to train $\pi^*_{0.6}$ (Physical Intelligence et al., 2025a), combines demonstrations, autonomous experience, and expert interventions to improve VLA policies through reinforcement learning. VLA Grounder (Shodiev et al., 2026) is particularly close to our architecture: it trains a guiding VLM with GRPO using a command prior derived from failures and sparse task-completion rewards, while keeping the downstream VLA frozen. Its formulation searches for grounded commands consistent with a supplied instruction and selects one command that remains fixed throughout a rollout. BABYSTEPS instead learns to revise task interpretations inferred from demonstrations across attempts through supervised fine-tuning of the upstream supervisor, while keeping the executor frozen.

Rationale distillation. Teacher explanations are established supervision signals: Distilling Step-by-Step (Hsieh et al., 2023) trains smaller models using language-model rationales alongside task labels. We study such supervision for task-interpretation revision conditioned on visual demonstrations and failed executions, constructing teacher-authored reflection–instruction targets from failure histories and evaluating whether repairs recover the demonstrated task.

3 METHOD

BABYSTEPS trains a vision-language supervisor to revise its interpretation of a demonstrated task using execution evidence. Prior work on multi-turn LLM agents identifies *belief traps*, where imperfect belief updates lead to repetitive, uninformative interactions (Zou et al., 2026). We study a related failure in robotic task interpretation: a supervisor can remain anchored to its initial belief, repeating or rephrasing an instruction without revising the inferred task interpretation based on the rollout result. Our supervision teaches the supervisor to reflect on the failed rollout, explain what to retain or revise, and produce a refined instruction for multi-turn attempts.

Figure 2 provides an overview of this framework, illustrating the test-time interaction loop alongside the offline supervision procedure.

3.1 PROBLEM FORMULATION

We consider demonstration-conditioned task execution with a vision-language supervisor q_θ and a frozen robot policy π_{VLA}. Each task τ has a canonical instruction g_τ and an environment success predicate C_τ, neither of which is supplied to the supervisor as a task specification at deployment. Instead, the supervisor receives a sequence D of timestamped reference-demonstration frames and an image o_0 of the robot's initial state s_0. Each evaluation instance is specified by a pair $(\texttt{demo_id}, \texttt{state_index})$: D is sampled from the official LIBERO human teleoperation dataset, while s_0 is selected from the benchmark's fixed evaluation initial states for the same task. The demonstration is a human teleoperated execution of the same task recorded from a different initial configuration, so object placements may differ from the robot's scene. All tasks use the same prompt and fixed instruction-style example; official object metadata is not provided. The supervisor must infer the intended goal and necessary subgoals from these observations.

An evaluation episode consists of this fixed demonstration–initial-state pair and up to $K = 5$ robot attempts. Let H_{k-1} denote the observable interaction history before attempt k, with $H_0 = \emptyset$, and

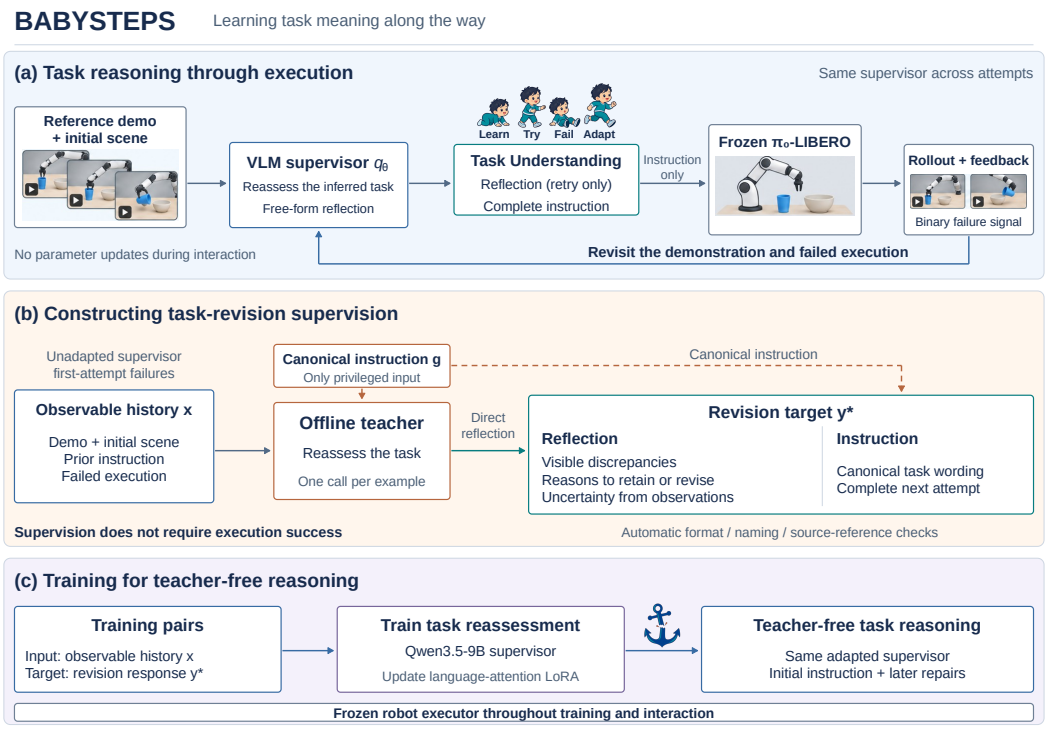

Figure 2: Overview of BABYSTEPS. (a) The supervisor revisits task interpretations using execution evidence while guiding a frozen VLA executor. (b) An offline teacher provides reflections paired with canonical instructions, without requiring execution success. (c) Training uses only observable history as input, enabling teacher-free reasoning across retries.

let $x_k = (D, o_0, H_{k-1})$. On the first attempt, q_θ outputs an instruction u_1. Following a failure, it generates a reflection r_k and a complete instruction u_k for the next attempt, $(r_k, u_k) \sim q_\theta(\cdot \mid x_k)$. The reflection reassesses the prior interpretation using the demonstration and execution evidence. Only u_k is passed to π_{VLA}; each instruction specifies the full task from s_0.

For an executed attempt, let ξ_k denote the rollout and $y_k = \max_t C_\tau(s_{k,t})$ its binary outcome, where $s_{k,t}$ is the environment state at step t. Execution stops as soon as the success predicate is satisfied, and the episode ends at the first successful attempt. We evaluate recovery within the attempt budget using

$$\text{Success@}\, k = \Pr(\exists j \le k : \; y_j = 1)\,, \qquad 1 \le k \le K. \tag{1}$$

Evaluation task identities are held out from supervisor training. The frozen π_0-LIBERO executor was trained on the evaluated LIBERO suites, so this split measures generalization by the supervisor.

To characterize *initial-belief anchoring*, let $z_\tau(u)$ denote the task-relevant objects, actions, and goal relations expressed by an instruction, ignoring paraphrases. Let u_{a} be the initial instruction, or the planted instruction in a controlled intervention. We operationalize full persistence of an incorrect initial interpretation at retry $k \ge 2$ as

$$A_k = \mathbf{1}[z_\tau(u_k) \equiv z_\tau(u_{\mathrm{a}}) \;\wedge\; z_\tau(u_{\mathrm{a}}) \not\equiv z_\tau(g_\tau)]\,. \tag{2}$$

Here z_τ is notation for semantic analysis, not an additional model component. Partial anchoring can retain only some erroneous task components. Verbatim instruction repetition is measured separately, since a repeated instruction may already express the correct task.

3.2 Execution and Feedback Protocol

At deployment, both the supervisor and executor parameters remain fixed. The first supervisor response contains only an `Instruction:` section. After an executed attempt fails, the supervisor

receives a binary failure notification with no explanation, the instruction actually executed, and sampled rollout frames. Its next response contains `Reflection:` followed by `Instruction:`.

Our main protocol, *recent visual history*, retains all prior text, including supervisor responses, executed instructions, and failure notifications, but keeps execution images only from the most recent failed attempt. The demonstration D and initial observation o_0 remain available throughout the episode. The *cumulative visual history* variant retains all prior execution images. Demonstration and rollout videos are sampled at 1 Hz with the final frame included; Appendix A.1 gives the complete input construction and prompts.

Every attempt restores the same initial state s_0 and reuses the same environment and policy-noise seeds under deterministic execution. The repair prompt explicitly states this reset protocol. A repeatability check on 40 episodes found matching success outcomes across both deterministic runs (Appendix A.1). We study recovery through instruction revision under these fixed execution conditions, although a wording change alone need not revise task meaning. Instructions that cannot be parsed, exceed the executor's 48-token language budget, or are truncated are classified as interface-invalid. They are not executed, receive $y_k = 0$, and terminate the episode unsuccessfully.

3.3 Supervision for task-interpretation revision

A failed rollout is ambiguous: the supervisor may have inferred the wrong task, or the executor may have failed to carry out a correct instruction. BABYSTEPS therefore supervises task revision using the canonical task instruction as a semantic reference, independently of whether executing that instruction succeeds. The aim is to teach the supervisor which parts of its interpretation to retain and which to reconsider.

We collect first-attempt failures from an unadapted supervisor and use each observable repair history x_i to construct a training example. A privileged teacher T receives x_i and the canonical instruction g_i, its only additional task information. The teacher is prompted to reassess the previous instruction against the demonstration and failed rollout, using g_i as the reference for task meaning, and directly writes a reflection $r_i^\star$ explaining what to retain or revise. We pair this reflection with the canonical instruction:

$$r_i^\star = T(x_i, g_i), \qquad y_i^\star = \mathrm{Format}(r_i^\star, g_i), \tag{3}$$

where Format uses the reflection–instruction response format in Section 3.2. The target instruction is exactly g_i. Privileged information enters through target construction; the supervisor conditions only on the observable history x_i. Targets are retained after textual validity checks, without requiring execution of g_i or a successful correction rollout.

We train the supervisor to generate the complete target response by minimizing

$$\mathcal{L}(\theta) = -\frac{1}{N_{\mathrm{tok}}} \sum_i \sum_{t=1}^{|y_i^\star|} \log q_\theta\big(y_{i,t}^\star \mid x_i, y_{i,<t}^\star\big)\,, \tag{4}$$

where $N_{\mathrm{tok}} = \sum_i |y_i^\star|$ is the total number of target tokens. All response tokens receive equal weight, while input tokens are excluded from the loss. We update LoRA adapters (Hu et al., 2021) in the supervisor's language-attention layers, keeping the remaining backbone parameters and the robot executor fixed.

Training examples supervise only the first repair response after a single failed attempt. At deployment, the adapted supervisor generates both the initial instruction and subsequent repairs, applying the learned revision behavior across retries without teacher access. Appendix A.2 provides the data construction, teacher prompts, validity checks, and training configuration.

4 Experiments

We evaluate whether BABYSTEPS improves execution on tasks held out from supervisor training, revises incorrect initial interpretations, and benefits from reflection supervision. Qwen3.5-9B is the main supervisor; a Qwen3.5-4B experiment examines transfer of the supervision across model sizes.

4.1 EXPERIMENTAL SETUP

We evaluate on LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-10 (Liu et al., 2023a). Each suite contributes six training tasks and four evaluation tasks, with the latter divided equally between DEV and TEST. Evaluation uses five demonstration–initial-state pairs per task, yielding 80 instances across 16 tasks held out from supervisor training. We use the official OpenPI `pi0_libero` checkpoint of π_0 (Black et al., 2024) as the frozen executor; it was trained on these suites.[1] For the end-to-end evaluation, all methods use the same instances, prompts, decoding settings, and robot execution seeds, with at most five attempts under the deterministic reset protocol in Section 3.2. We report Success@k using recent visual history and compare cumulative visual history in Section 4.3.

Baselines. *Zero-shot Qwen3.5-9B* uses the official Qwen3.5-9B checkpoint (Qwen Team, 2026) without an adapter. The model-size transfer experiment uses the corresponding official Qwen3.5-4B checkpoint as its zero-shot baseline.[2] *Instruction-only SFT* uses the same 67 training instances and 12 validation instances as BABYSTEPS, but learns to predict the canonical instruction from the demonstration and initial scene alone. Its training inputs contain no failed attempts or execution frames, and its targets contain no reflections. At evaluation, it receives the same repair prompts and generates reflections on retries. This baseline tests how much performance can be obtained from canonical-instruction supervision alone. *Neutral-reflection SFT* uses the same repair inputs, canonical instruction targets, and training recipe as BABYSTEPS, but replaces every teacher-authored reflection with a fixed generic statement acknowledging failure and proposing to restate the task. It provides a comparison with generic reflection supervision and is evaluated in the controlled analyses below.

Training and model selection. The adapted supervisors use fresh LoRA adapters with rank 8 and scaling factor 16, a learning rate of 10^{-4}, and at most eight training epochs. Checkpoints are selected by normalized instruction exact match on 12 validation examples from the training-task pool. This validation set is distinct from DEV. Prompts and the training recipe were fixed before the present evaluation; DEV and TEST outcomes were not used for checkpoint selection. We report their combined results in the main table and separate results in Appendix B. All reported BABYSTEPS results use reflection supervision without policy optimization.

4.2 MAIN RESULTS

Table 1 shows that BABYSTEPS with Qwen3.5-9B reaches 77.5% Success@5, exceeding Zero-shot by 30.0 percentage points and Instruction-only SFT by 10.0 points. Success increases from 44/80 instances on the first attempt to 62/80 within five attempts. Although training supervises only repair responses, first-attempt success also improves to 55.0%, compared with 50.0% for Instruction-only SFT and 37.5% for Zero-shot. On TEST alone, Success@5 is 70.0% for BABYSTEPS, 50.0% for Instruction-only SFT, and 52.5% for Zero-shot.

Transfer to a smaller supervisor. We train Qwen3.5-4B on the same 67/12 training and validation examples, without collecting failures from the 4B model. Its repair supervision therefore comes from histories generated by a different supervisor. BABYSTEPS-4B also reaches 77.5% Success@5, compared with 57.5% for Zero-shot and 67.5% for Instruction-only SFT. Its first-attempt success is lower than Instruction-only SFT (36.3% versus 40.0%), but it leads from the second attempt onward. The final advantage therefore does not depend on starting with a stronger first instruction.

Repair from shared failure histories. Table 2 compares all four 9B variants on the same 25 DEV first-attempt failures from Zero-shot. These histories were collected under the default nondeterministic execution setting, and every history is retained. Each method generates eight independent repairs per history, yielding 200 rollouts per method, with the original failure frames preserved as inputs. Repair execution uses deterministic settings with the same initial state and fixed seeds. Replaying each unchanged initial instruction once under the later deterministic setting succeeds on

[1] Official OpenPI implementation and checkpoint configuration: `Physical-Intelligence/openpi`.
[2] Official checkpoints: `Qwen/Qwen3.5-9B` and `Qwen/Qwen3.5-4B`.

Table 1: Success@k (%) on 80 evaluation instances with recent visual history. Adapted 4B models use the same data and training recipe as their 9B counterparts, including 9B-generated repair histories for BABYSTEPS. Best results within each backbone are bold.

Backbone	Method	Success@k (%)				
		@1	@2	@3	@4	@5
Qwen3.5-9B	Zero-shot	37.5	46.3	47.5	47.5	47.5
	Instruction-only SFT	50.0	56.3	63.8	67.5	67.5
	BABYSTEPS	**55.0**	**67.5**	**70.0**	**72.5**	**77.5**
Qwen3.5-4B	Zero-shot	30.0	46.3	51.3	53.8	57.5
	Instruction-only SFT	**40.0**	60.0	63.8	66.3	67.5
	BABYSTEPS	36.3	**63.8**	**70.0**	**75.0**	**77.5**

Table 2: Per-sample repair success with Qwen3.5-9B on all 25 shared DEV first-attempt failure histories (eight independently sampled repairs per history). All methods receive the same recorded nondeterministic rollouts; repairs are evaluated with deterministic robot execution. Each method has 200 repair rollouts. Replay evaluates each unchanged initial instruction once in the later deterministic mode; its three successes remain in the repair pool.

Method	Successful rollouts	Success (%)
Zero-shot	46/200	23.0
Instruction-only SFT	54/200	27.0
Neutral-reflection SFT	59/200	29.5
BABYSTEPS	104/200	52.0
Replay initial instruction	3/25	12.0

3/25 histories; these histories remain in the repair pool. We report *per-sample repair success*, not Success@8 over sequential retries (Appendix B.1).

4.3 Ablation studies

Reflection supervision. Neutral-reflection SFT uses the same failed histories, canonical instruction targets, and training recipe as BABYSTEPS, but replaces teacher-authored reflections with a fixed generic statement acknowledging failure and restating the task. The shared-history comparison in Section 4.2 holds the repair inputs fixed across these target schemes. Appendix A.5 discusses the associated differences in target length and token weighting.

Amount of visual history. Table 3 compares retaining only the most recent failed rollout with retaining all failed-rollout images; both protocols preserve the full text history. BABYSTEPS reaches 77.5% with recent visual history and 73.8% with cumulative visual history, outperforming Instruction-only SFT by 10.0 percentage points under both protocols. Recent visual history was selected for primary reporting after both protocols had been evaluated on all 80 instances. Reporting both shows that the method ranking is preserved across this choice.

Human preference evaluation. Two annotators compared repair responses (reflections and revised instructions) for the same 25 DEV failures, yielding 50 judgments with model identities and revised execution outcomes hidden. BABYSTEPS was selected in 30/50 judgments (60%), compared with 10/50 (20%) each for Zero-shot and Instruction-only SFT. Agreement was low (48%, Cohen's $\kappa = 0.09$), so we treat this as supporting evidence of response preference; Appendix D gives the protocol and limitations.

Table 3: Success@5 (%) for Qwen3.5-9B under two visual-history protocols on the same 80 instances. Both retain all prior text.

Method	Recent visual history	Cumulative visual history
Zero-shot	47.5	50.0
Instruction-only SFT	67.5	63.8
BABYSTEPS	**77.5**	**73.8**

Table 4: Outcomes of one repair after an implanted incorrect instruction (Qwen3.5-9B). Counts are out of 60 task pairings from 20 evaluation instances. *True task included* permits additional planted components. Rows overlap; mutually exclusive semantic categories are reported in Table 9.

Outcome	Zero-shot	Instruction-only SFT	Neutral-reflection SFT	BABYSTEPS
Planted task only	42	33	0	0
Any planted component	49	55	5	12
Verbatim repetition	29	22	0	0
True task only	1	1	8	**20**
True task included	1	3	8	**22**
Environment success	1	3	8	**19**

5 Analysis

We examine how recovery changes the task interpretation expressed in the supervisor's instructions. The analysis separates persistence of an incorrect interpretation, revision toward the demonstrated goal, and successful execution.

5.1 Persistence of incorrect task interpretations

During natural retries, Zero-shot Qwen3.5-9B repeats its immediately preceding instruction verbatim on 93/177 repair turns (52.5%), compared with 38/130 (29.2%) for Instruction-only SFT and 9/108 (8.3%) for BABYSTEPS. These verbatim repetitions leave the expressed task interpretation unchanged. The corresponding rates for the 4B variants are 24.6%, 19.9%, and 5.6%. These rates describe the histories induced by each supervisor; repetition alone does not establish that the repeated instruction expresses an incorrect task.

To compare semantic revision under identical histories, we use the controlled incorrect-instruction intervention on 20 LIBERO-Goal instances spanning four tasks. For each instance, we insert each of the other three tasks' canonical instructions as the supervisor's initial answer, producing 60 task pairings. The frozen executor runs each planted instruction, and all 60 attempts fail the true task's success condition. Each of the four 9B supervisors then receives the same demonstration, initial scene, planted answer, and failed-rollout frames, generates one repair, and executes it from the original scene. Semantic categories are assigned to the executed instructions using task-specific rules followed by non-blind model-assisted review, with human spot-checks on a subset of outputs. Execution success is measured separately by the environment predicate. The intervention and annotation protocol are detailed in Appendix C.

Table 4 shows that Zero-shot Qwen3.5-9B and Instruction-only SFT preserve at least one planted task component in 49/60 and 55/60 repairs, respectively. They express only the planted task in 42 and 33 cases. Both Neutral-reflection SFT and BABYSTEPS eliminate this complete persistence in the probe, while some repaired instructions still retain planted components. The neutral control retains such components less often than BABYSTEPS (5/60 versus 12/60), making the direction of revision essential to interpreting recovery.

5.2 Revision toward the demonstrated task

BABYSTEPS expresses the true task without planted components in 20/60 repairs, compared with 8/60 for Neutral-reflection SFT, and completes the task in 19/60 versus 8/60 cases. Although the neutral control retains fewer planted components, it more often replaces them with another incorrect interpretation. Thus, abandoning more of an incorrect instruction does not necessarily yield a more accurate revision; the advantage of BABYSTEPS is more frequent recovery toward the demonstrated task.

Semantic revision and execution success remain distinct. Of the instructions classified as expressing only the true task, 18/20 succeed for BABYSTEPS and 6/8 for the neutral control. Conversely, two neutral-control instructions succeed despite lacking the complete true-task description. Instruction semantics and environment success therefore provide complementary evidence about recovery.

5.3 Residual errors

Residual errors include incomplete revision, in which repaired instructions retain planted components, and incorrect reinterpretation, in which they abandon the planted task but choose a different wrong goal. Some repairs combine true-task and planted components; others identify the relevant object or action but specify an incorrect goal. These patterns help explain why frequent instruction changes do not necessarily recover the demonstrated task. Appendix C.2 gives the category breakdown, and Appendix C.3 reports natural performance on the same episodes without implantation.

6 Conclusion

We introduced BABYSTEPS, a framework that learns to revise task interpretations inferred from demonstrations after execution failure. By pairing teacher-authored reflections with canonical instructions, our method supervises semantic revision without requiring successful correction rollouts. The trained supervisor guides a frozen VLA executor using only demonstrations and observable interaction history at deployment. On LIBERO tasks held out from supervisor training, BABYSTEPS achieves 77.5% Success@5 with Qwen3.5-9B, improving over Zero-shot and Instruction-only SFT by 30 and 10 percentage points. Controlled interventions further show more frequent recovery toward the demonstrated task than a neutral-reflection control. These findings support task-interpretation revision as a component of recovery and highlight the distinction between abandoning an incorrect instruction and recovering the intended goal.

Our evaluation is limited to simulated LIBERO tasks with a single frozen executor, and target construction relies on canonical task instructions. Extending this approach to real robots, broader task distributions, and supervision without canonical instructions remains an open direction.

AI use statement

We use gpt-5.6-sol to author reflection targets for offline supervisor training (Appendix A.2) and Claude to assist semantic annotation of the implanted-history probe (Appendix C.2). These probe annotations include human spot-checks on a subset of outputs, rather than exhaustive human review. We also use generative AI to assist manuscript wording and formatting. The authors take responsibility for the final manuscript and reported results.

Ethics statement

Robotic evaluations use LIBERO simulation and existing demonstrations collected through human teleoperation. The supplementary preference study collects judgments from two annotators on recorded simulated episodes and repair responses, without physical robot interaction. Our evaluation does not establish safe deployment on physical robots: a revised instruction can remain incorrect, and a semantically correct instruction can still fail during execution. Semantic annotations use non-blind model-assisted review with partial human checking; environment success is measured separately. These limitations should be considered when interpreting the findings or extending the method to physical systems.

Reproducibility statement

Section 4.1 identifies the benchmark, official supervisor checkpoints, frozen executor, and evaluation split. Appendix A describes visual inputs, interaction prompts, teacher-target construction and filtering, LoRA training, checkpoint selection, and the neutral-reflection control. Appendix B reports complete evaluation counts and the shared-failure protocol, while Appendix C details the intervention, decoding settings, semantic annotation, and natural-performance comparison. Appendix D documents the human preference study's cases, presentation, question, and results.

References

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022. URL https://arxiv.org/abs/2204.01691.

Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. URL https://arxiv.org/abs/2410.24164.

Dongzhou Cheng, Taoran Yi, Ye Fang, Xingwu Zhang, Fan Feng, Yixuan Li, Gengxiong Zhuang, Rongze Wang, Shuai Yang, Wei Song, Weizhi Xue, Minyan Wu, Jie Gui, Jiaqi Wang, and Tong Wu. In-context robot learning with VLM agents. *arXiv preprint arXiv:2609.19138*, 2026. URL https://arxiv.org/abs/2609.19138.

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 Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 45493–45517, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/70a06501001e1820fd1eb9ee821302d2-Paper-Conference.pdf.

Letian Fu, Huang Huang, Gaurav Datta, Lawrence Yunliang Chen, William Chung-Ho Panitch, Fangchen Liu, Hui Li, and Ken Goldberg. In-context imitation learning via next-token prediction. *arXiv preprint arXiv:2408.15980*, 2024. URL https://arxiv.org/abs/2408.15980.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling Step-by-Step! Outperforming Larger Language Models with Less Training Data and Smaller Model Sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 8003–8017, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.507. URL https://aclanthology.org/2023.findings-acl.507/.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. URL https://arxiv.org/abs/2106.09685.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. VoxPoser: Composable 3D value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023. URL https://arxiv.org/abs/2307.05973.

Yunfan Jiang, Yevgen Chebotar, Ruijie Zheng, Fengyuan Hu, Yunhao Ge, Jimmy Wu, Tianyuan Dai, Scott Reed, Li Fei-Fei, Yuke Zhu, and Linxi "Jim" Fan. RoboTTT: Context scaling for robot policies. *arXiv preprint arXiv:2607.15275*, 2026. URL https://arxiv.org/abs/2607.15275.

Ruofei Ju, Xinrui Wang, Xin Ding, Yifan Yang, Hao Wu, Shiqi Jiang, Qianxi Zhang, Hao Wen, Xiangyu Li, Weijun Wang, Kun Li, Yunxin Liu, Haipeng Dai, Wei Wang, and Ting Cao. Embodiskill: Skill-aware reflection for self-evolving embodied agents, 2026. URL https://arxiv.org/abs/2605.10332.

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

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. URL https://arxiv.org/abs/2306.03310.

Jijia Liu, Feng Gao, Bingwen Wei, Xinlei Chen, Qingmin Liao, Yi Wu, Chao Yu, and Yu Wang. What can RL bring to VLA generalization? an empirical study. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2505.19789.

Zeyi Liu, Arpit Bahety, and Shuran Song. REFLECT: Summarizing robot experiences for failure explanation and correction. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 3468–3484. PMLR, 2023b. URL https://proceedings.mlr.press/v229/liu23g.html.

Zhen Luo, Yixuan Yang, Yanfu Zhang, and Feng Zheng. Roboreflect: A robotic reflective reasoning framework for grasping ambiguous-condition objects, 2025. URL https://arxiv.org/abs/2501.09307.

Physical Intelligence, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Jared DiCarlo, Danny Driess, Michael Equi, Adnan Esmail, Yunhao Fang, Chelsea Finn, Catherine Glossop, Thomas Godden, Ivan Goryachev, Lachy Groom, Hunter Hancock, Karol Hausman, Gashon Hussein, Brian Ichter, Szymon Jakubczak, Rowan Jen, Tim Jones, Ben Katz, Liyiming Ke, Chandra Kuchi, Marinda Lamb, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Yao Lu, Vishnu Mano, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Charvi Sharma, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, Will Stoeckle, Alex Swerdlow, James Tanner, Marcel Torne, Quan Vuong, Anna Walling, Haohuan Wang, Blake Williams, Sukwon Yoo, Lili Yu, Ury Zhilinsky, and Zhiyuan Zhou. $\pi^*_{0.6}$: a VLA that learns from experience. *arXiv preprint arXiv:2511.14759*, 2025a. URL https://arxiv.org/abs/2511.14759.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025b. URL https://arxiv.org/abs/2504.16054.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Lucy Xiaoyang Shi, Brian Ichter, Michael Robert Equi, Liyiming Ke, Karl Pertsch, Quan Vuong, James Tanner, Anna Walling, Haohuan Wang, Niccolo Fusai, Adrian Li-Bell, Danny Driess,

Lachy Groom, Sergey Levine, and Chelsea Finn. Hi Robot: Open-ended instruction following with hierarchical vision-language-action models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 54919–54933. PMLR, 2025. URL https://proceedings.mlr.press/v267/shi25d.html.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

Damir Shodiev, Aleksei Staroverov, Nikita Kachaev, Alexey K. Kovalev, and Aleksandr I. Panov. VLA Grounder: Language-conditioning space optimization for black-box VLA models. *arXiv preprint arXiv:2607.04517*, 2026. URL https://arxiv.org/abs/2607.04517.

Beichen Wang, Juexiao Zhang, Shuwen Dong, Irving Fang, and Chen Feng. Vlm see, robot do: Human demo video to robot action plan via vision language model, 2025. URL https://arxiv.org/abs/2410.08792.

Yuki Wang, Gonzalo Gonzalez-Pumariega, Yash Sharma, and Sanjiban Choudhury. Demo2code: From summarizing demonstrations to synthesizing code via extended chain-of-thought. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 14848–14956. Curran Associates, Inc., 2023. doi: 10.52202/075280-0655. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/30699996ff411d48903c9752b782a5c1-Paper-Conference.pdf.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183. PMLR, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

Deyu Zou, Yongqiang Chen, Jianxiang Wang, Garry YANG, Mufei Li, Qing Da, James Cheng, Pan Li, and Yu Gong. Reducing belief deviation in reinforcement learning for active reasoning of LLM agents. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=r8hzDA3pUY.

A Method Implementation Details

A.1 Visual inputs and interaction protocol

Reference demonstrations are human-teleoperated LIBERO trajectories for the same task as the robot episode, but with different initial object arrangements. The input D contains 5–25 timestamped frames sampled at approximately 1 Hz, always including the final frame. Each failed rollout contributes 13–28 chronological frames including its final frame. Demonstration and rollout frames are resized to 512×512. The supervisor also receives one image of the original robot scene.

Every first-attempt prompt includes the same sentence: "Example of the expected instruction style: put the bowl on the plate." This example coincides with the official instruction of training task

libero_goal_008, but is supplied identically for all tasks. No task-specific official instruction, simulator success predicates, or official object-name metadata is supplied in the supervisor's inference context.

The first response contains only an Instruction: section. Each subsequent response contains a Reflection: section followed by an Instruction: section. The prompt requests a reflection of at most 80 words and a concise instruction, preferably under 25 words, subject to the executor's 48-token language limit. Only the instruction is sent to the frozen pi0_libero executor. A rollout attempts the entire task; it is not an individual subtask. The control-step budget is suite dependent, taking values of 220, 280, 300, or 520. Responses that cannot be parsed, are truncated, or contain an instruction exceeding 48 tokens are not executed and count as failed attempts. No such invalid responses occurred in the reported runs.

After failure, the supervisor receives the message "FAIL. The environment success condition was not satisfied.", the instruction actually sent, and chronological rollout frames including the final frame. In the end-to-end multiround evaluation, every retry restores the same initial state and reuses the same environment and robot-policy noise seeds under deterministic execution. In a repeatability check that ran the same 40 episodes twice in deterministic mode, action sequences matched exactly in 39/40 episodes and success outcomes matched in all 40/40. The fixed-failure repair comparison instead uses initial failures collected before deterministic execution was enabled; Appendix B.1 distinguishes that collection from subsequent deterministic replay.

A.2 Teacher targets and data selection

The unadapted Qwen3.5-9B supervisor produces one first-attempt rollout per training row. Of 115 failed attempts, deterministic selection with fixed quotas retains 84 for teacher annotation. The offline teacher, gpt-5.6-sol, receives the original student conversation in the same order and with the same images, plus the official instruction string. No success predicates, BDDL specification, or object list is supplied. Each selected history receives one teacher call, with no retries or student rewriting.

The teacher returns a reflection, a category (meaning or wording), and an auxiliary rationale. The category and rationale are archived but excluded from training. The prompt requests that reflections rely only on student-visible evidence, separate observations from possible explanations, preserve supported goals, and express uncertainty when the observations are inconclusive. This is a prompting requirement, not a verified property of the generated reflections.

An automatic lint filter rejects reflections containing prohibited references to privileged information, benchmark or dataset names, underscore identifiers, line breaks, or more than 100 words. Prohibited terms include "official," "label," "teacher," "ground truth," and "correct instruction." Canonical object names are permitted only as parenthetical aliases immediately following visible descriptions, such as "the blue-and-yellow can (alphabet soup)." These aliases allow canonical object names from the privileged instruction to enter reflection targets; the student input does not include that privileged instruction. The prohibited-reference checks restrict references to privileged sources, not all information derived from them. The filter only rejects outputs; it does not rewrite them or assess semantic correctness or evidential support. It rejects five examples: four violate the naming rule, and one is a false positive because "blue-labeled" matches the prohibited term "label."

The 79 accepted examples comprise 48 teacher-labeled meaning cases, whose initial instructions misstate the task, and 31 wording cases, whose initial instructions express the same task in different words. None of the 79 initial instructions exactly matches the canonical instruction. Consequently, every target changes the instruction text: retention concerns the task interpretation described in the reflection, and no example supervises verbatim instruction retention. We divide them into 67 training examples and 12 validation examples. We do not filter by execution success under the canonical instruction. Among the accepted examples, corresponding canonical-instruction rollouts succeed in 37 cases and fail in one; no corresponding rollout is available for the remaining 41.

A.3 Training configuration

Each input contains the system message, reference demonstration, initial scene, first-attempt request, the Zero-shot supervisor's first response, the failure notification with the instruction actually

Table 5: Training configuration. Only language-model attention LoRA parameters are updated.

Supervisor	Qwen3.5-9B
Executor	Frozen π_0-LIBERO
LoRA rank / scaling	$r = 8$, $\alpha = 16$
Learning rate	10^{-4}
Effective batch size	8 examples
Maximum training epochs	8
Training / validation examples	67 / 12
Checkpoint selection	Normalized validation instruction exact match
Selected checkpoint	Epoch 7

executed, failed-rollout frames, and the repair request. Its serialization matches the corresponding repair input used at inference. It contains neither a student reflection draft nor teacher feedback, the canonical instruction, or the teacher's auxiliary annotations.

The target consists of the teacher-written reflection, the official instruction, section-formatting tokens, and the end-of-sequence token. All target tokens are supervised, while history tokens receive zero loss. The loss is normalized by the total number of supervised tokens across the effective batch of eight examples. Reflection and instruction tokens have equal weights; reflection text accounts for approximately 82% of supervised tokens.

The vision tower, language-model MLPs, and all other non-LoRA parameters remain frozen. We select the checkpoint that maximizes the number of validation examples whose generated instruction matches the official instruction after normalization, breaking ties in favor of the earliest epoch; the selected checkpoint is epoch 7. Training uses only the supervised objective in Equation 4; no reinforcement-learning stage is used.

A.4 Visual history and evaluation protocols

Recent visual history, our default for the main results, retains only the latest failed rollout's frames, removing older execution frames from the third attempt onward. *Cumulative visual history* retains execution frames from every previous attempt. Both settings retain the reference demonstration, initial scene, and all previous textual responses, executed instructions, and failure notifications.

The end-to-end and fixed-failure evaluation protocols are described in Sections 4.1 and 4.2. In particular, initial responses are generated by each method's own supervisor in the end-to-end evaluation, whereas the fixed-failure diagnostic shares the same Zero-shot supervisor responses and failed-rollout contexts across methods.

A.5 Neutral-reflection control

The neutral-reflection control uses the same 67 training rows and 12 validation rows as BABYSTEPS. The data-construction script checks equality of the student messages and encoded prompts, including image-tensor hashes and token types. Both models use the same canonical instruction targets and initialize a new LoRA adapter on Qwen3.5-9B. The only change to the training targets is replacement of each teacher reflection with the same sentence: "The robot moved but the final frames do not match the demonstration. I will restate the whole demonstrated task plainly." The instruction and output format are unchanged.

Both runs use LoRA rank 8 and scaling 16 on 152 language-model attention projections, a constant learning rate of 10^{-4}, microbatch size 1, gradient accumulation over eight examples, seed 11401, and eight epochs comprising 72 optimizer updates. The loss supervises the complete response through the end-of-sequence token. The shared token-normalized objective means that different reflection lengths also change the relative contribution of instruction tokens. This control therefore compares reflection-target schemes, rather than isolating semantic content at a fixed target length.

Both runs select the epoch with the highest normalized validation instruction exact-match count, taking the earliest epoch in a tie. Across epochs 1–8, the counts are $(0, 0, 0, 0, 3, 3, 8, 4)$ for

Table 6: Additional Success@k curves (%). The 9B recent-history setting is the main comparison in Table 1. The 4B models were evaluated only with recent visual history.

Size	History	Supervisor	@1	@2	@3	@4	@5
9B	Cumulative	Zero-shot	37.50	46.25	48.75	50.00	50.00
9B	Cumulative	Instruction-only SFT	50.00	56.25	61.25	63.75	63.75
9B	Cumulative	BABYSTEPS	55.00	67.50	70.00	73.75	73.75
4B	Recent	Zero-shot	30.00	46.25	51.25	53.75	57.50
4B	Recent	Instruction-only SFT	40.00	60.00	63.75	66.25	67.50
4B	Recent	BABYSTEPS	36.25	63.75	70.00	75.00	77.50

Table 7: Complete Success@5 results. Counts are out of 80 overall and 40 per split; rates and task-clustered 95% confidence limits are percentages. The first three rows are the main setting.

Size	History	Supervisor	Count	Rate	95% CI	DEV	TEST
9B	Recent	Zero-shot	38	47.50	[31.25, 62.50]	17	21
9B	Recent	Instruction-only SFT	54	67.50	[51.25, 82.50]	34	20
9B	Recent	BABYSTEPS	62	77.50	[62.50, 88.75]	34	28
9B	Cumulative	Zero-shot	40	50.00	[32.50, 66.25]	20	20
9B	Cumulative	Instruction-only SFT	51	63.75	[46.25, 80.00]	31	20
9B	Cumulative	BABYSTEPS	59	73.75	[58.75, 86.25]	31	28
4B	Recent	Zero-shot	46	57.50	[41.25, 73.75]	23	23
4B	Recent	Instruction-only SFT	54	67.50	[48.75, 85.00]	30	24
4B	Recent	BABYSTEPS	62	77.50	[61.25, 91.25]	36	26

BABYSTEPS and $(0, 1, 5, 4, 9, 8, 9, 10)$ for the control, selecting epochs 7 and 8, respectively. The Instruction-only SFT model uses epoch 5. The neutral control is evaluated in the fixed-failure and implanted-history diagnostics; no full Success@k evaluation is available for this control.

B Complete Evaluation Results

All multiround evaluations use the same pool of 80 episodes, comprising 40 DEV and 40 TEST episodes from 16 tasks held out from supervisor adaptation. Each suite contributes 20 episodes. The main 9B recent-history curves appear in Table 1; Table 6 gives the remaining curves. Table 7 reports every Success@5 count, confidence interval, and split breakdown, and Table 8 reports suite counts. Confidence intervals use a task-clustered bootstrap with 10,000 resamples and seed 20260924. All percentages are calculated from the integer counts, preserving quarter-percentage-point values.

B.1 Shared failure histories for the repair comparison

Table 2 uses all 25 Zero-shot DEV first-attempt failures collected before deterministic π_0 execution was enabled. Their `physical.json` files record no π_0 flag; collection used the default nondeterministic setting. Each of the four models samples eight repairs per original history by varying the supervisor seed. Repairs restore the initial state with fixed environment and policy-noise seeds under deterministic XLA execution. All 200 repairs per method enter per-sample success; replay does not filter the pool.

Replaying the unchanged instructions in the later deterministic mode succeeds on 3/25 episodes: `libero_goal_002_s28_demo_4`, `libero_object_002_s30_demo_0`, and `libero_object_002_s35_demo_44`. All three remain included. These outcomes compare execution modes; repeatability within deterministic mode was tested separately (Appendix A.1).

C Implanted-History Probe

C.1 Instances, intervention, and matched evaluation

The probe includes all 20 LIBERO-Goal episodes in the main evaluation pool. Its four tasks are `goal_002` (placing the wine bottle on the cabinet top) and `goal_007` (turning on the stove) from DEV, and `goal_000` (opening the middle drawer) and `goal_005` (pushing the plate to the front

Table 8: Successful episodes within five attempts, out of 20 per suite. Columns follow LIBERO-10, LIBERO-Goal, LIBERO-Object, and LIBERO-Spatial.

Size	History	Supervisor	10	Goal	Object	Spatial
9B	Recent	Zero-shot	10	4	15	9
9B	Recent	Instruction-only SFT	16	10	9	19
9B	Recent	BABYSTEPS	17	9	17	19
9B	Cumulative	Zero-shot	10	4	15	11
9B	Cumulative	Instruction-only SFT	13	10	9	19
9B	Cumulative	BABYSTEPS	17	9	13	20
4B	Recent	Zero-shot	14	4	16	12
4B	Recent	Instruction-only SFT	18	7	10	19
4B	Recent	BABYSTEPS	18	10	17	17

of the stove) from TEST. Each task contributes five episodes. None of these four tasks occurs in supervisor training, which uses the other six LIBERO-Goal tasks.

For every episode, we implant each of the other three task instructions as a previous assistant answer. This produces 12 ordered true-task/implanted-task combinations with five episodes each. All 60 pairs are specified before execution and executed without filtering; all implanted attempts fail the true task. Each model receives the implanted answer and its actual failed rollout, then samples a single repair. The original scene is restored before executing the repair.

The four 9B models have matching serialized-message hashes for all 60 pairs and use the same per-pair repair seed. All use Hugging Face Transformers with temperature 0.7, top-p 0.8, top-k 20, presence penalty 1.5, a maximum of 768 new tokens, and thinking disabled. All 240 generated responses pass the strict parser, including the Instruction-only SFT responses; no permissive parsing fallback is used. The executor is `pi0_libero` with deterministic XLA settings, the same per-episode noise seed, and a 300-step budget. These are evaluation settings; no reinforcement-learning rollout sampling is used for these results.

C.2 Semantic annotation

Annotations apply only to the instruction actually sent to the robot. A reflection that mentions the implanted task while discussing the failure is not counted as retaining that task. Task-specific keyword rules first assign a coarse class based on whether the instruction mentions true-task components, implanted-task components, both, or neither. Separate rules assign one of the eight mutually exclusive semantic categories in Table 9.

Claude reviews the rule-based labels with access to model identity, true and implanted task labels, and execution outcome, making four coarse-class and three semantic-category overrides. We retain both rule-based and reviewed labels. The authors spot-check a subset of outputs and labels. This is non-blind model-assisted annotation with partial human checking, not exhaustive or blinded human assessment; environment success is reported separately.

Table 9 distinguishes complete true-task descriptions, with or without an extra implanted component, from partial mixtures that do not fully express the true task. Other categories distinguish complete or partial persistence of the implanted task, substitution of a third task, and true-task objects or actions paired with an incorrect goal. Outputs outside these categories are unclassified.

C.3 Natural performance and diagnostic overlap

Without implanted histories, Success@1→Success@5 counts on the main evaluation pool's 20 Goal episodes are Zero-shot $4 \rightarrow 4$, Instruction-only SFT $9 \rightarrow 10$, and BABYSTEPS $9 \rightarrow 9$ under both visual-history protocols; the canonical instruction succeeds on 19/20 in one attempt. Neutral-reflection SFT has no corresponding multiround evaluation. Six episodes (one `goal_002`, five `goal_007`) also belong to the same 25-history failure pool used for the repair comparison in Table 2.

Table 9: Mutually exclusive semantic categories for the implanted-history probe. Each column contains 60 repairs; parentheses give environment successes within that category. Entries without parentheses have zero successes.

Instruction category	Zero-shot	Instruction-only SFT	Neutral-reflection SFT	BABYSTEPS
True task only	1 (1)	1 (1)	8 (6)	20 (18)
True task + implanted component	0	2 (1)	0	2 (1)
Partial mixture	2	7	4 (1)	9
Implanted task only	42	33	0	0
Implanted component only	5	13	1	1
Third task	10	0	18	5
True object/action, wrong goal	0	4 (1)	24 (1)	18
Other / unclassified	0	0	5	5
Environment successes	1	3	8	19

D HUMAN PREFERENCE EVALUATION

Cases and presentation. JL and cw each evaluated the 25 original Zero-shot DEV failures used in Table 2. For each case, we used sample 0 from Zero-shot, Instruction-only SFT, and BABYSTEPS, generated from identical failure inputs.[3] Responses were labeled A/B/C, with method identities hidden and letter assignments randomized per item.

Annotators saw the demonstration video, initial scene, failed-attempt video and key frames, the instruction issued for that attempt, and each candidate's reflection and revised instruction. They could expand the task goal; revised execution outcomes were hidden. The prompt was: *Which reflection makes the most sense given the demonstration and the failed attempt?* Options were A, B, C, or *No clear difference*. Because both reflection and instruction were displayed, we interpret ratings as preferences over complete repair responses.

Preference and agreement. Table 10 reports 50 judgments over 25 cases, not 50 independent cases. Neither annotator selected *No clear difference*. Both selected BABYSTEPS most often, but agreed on only 12/25 choices (48%; Cohen's $\kappa = 0.09$).

Table 10: Human preferences over repair responses: 25 cases per annotator, with 50 judgments in total.

Annotator	Zero-shot	Instruction-only SFT	BABYSTEPS
JL	7	4	14
cw	3	6	16
Total	10/50 (20%)	10/50 (20%)	30/50 (60%)

Descriptive execution subsets. BABYSTEPS was selected in 15/22 judgments when its revised attempt failed and 15/28 when it succeeded. Hiding execution outcomes prevents direct outcome-based voting but establishes neither independence between preference and execution success nor semantic correctness of preferred responses.

Scope and limitations. With two annotators, 25 cases, and low agreement, this study provides supporting evidence of response preference. Visible revised instructions and expandable task goals allow instruction correctness, canonical wording, and overall presentation to influence choices, so the study does not isolate reflection quality. Its cases and question differ from the 60-pair implanted-history probe; these preferences neither validate that probe's semantic labels nor replace a blinded semantic review (Appendix C.2).

[3] All three original nondeterministic failures that succeed under later deterministic replay remain included: ratings concern the displayed original failure (Appendix B.1).