KNOWING IS NOT ACTING IN VLA MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language-action (VLA) policies can identify an object from an indirect description yet fail to select it when acting. We study this knowledge-action gap with KnowAct, which pairs literal and knowledge-based references to the target and an alternative object in balanced manipulation scenes. Its contrastive knowledge utilization (KU) measures knowledge-conditioned selection relative to literal control on a fixed set screened for reported knowledge and initial control. We also introduce Referential Self-Distillation (RSD), which trains direct execution against an EMA teacher conditioned on a translated object name. We evaluate GR00T N1.7 and $\pi_{0.5}$ in RoboTwin 2.0. On GR00T, RSD reaches KU 0.311 versus 0.210 for a collection-protocol-matched DAgger control; the paired item-bootstrap difference is 0.101 $[0.067, 0.135]$. On $\pi_{0.5}$, RSD reaches 0.345 versus 0.258 for DAgger, a paired difference of 0.087 $[0.048, 0.126]$. Among 157 GR00T items, knowledge contrast improves on 110, worsens on 35, and is unchanged on 12. The effect is positive in each evaluated split; knowledge injection, readout probes, activation patching, and control experiments further delimit the finding. The item-bootstrap intervals condition on three trained runs and do not quantify retraining uncertainty.

1 INTRODUCTION

Consider a tabletop with cream cheese, peanut butter, strawberry jam, and honey, and the instruction "pick up the spread made from milk." A VLA policy that grasps the peanut butter has failed, but the failure has several possible sources: the policy may lack the fact that cream cheese is made from milk, may know the fact and fail to connect it to the object in the image, or may identify the object and still be unable to grasp it. Replacing the description with "pick up the cream cheese" removes the knowledge step while keeping the scene and the motor task, and the difference between the two requests is informative only on items where the model answers the knowledge question and executes the literal instruction.

VLA models build on vision-language backbones so that knowledge acquired in pretraining can inform control (Brohan et al., 2023; Kim et al., 2024; Black et al., 2025; Physical Intelligence et al., 2025; Bjorck et al., 2025), and recent benchmarks show that this transfer is uneven (Kachaev et al., 2026; Kim et al., 2026). Both rest on aggregate success, which mixes knowledge, reference ambiguity, and control, and neither measures how much of the knowledge a frozen backbone can report reaches the action head that reads it. Language models show an analogous gap, where a recalled fact is not always usable in a downstream computation (Allen-Zhu & Li, 2023; Berglund et al., 2024; Balesni et al., 2024).

We make this gap measurable with KnowAct, a four-object selection protocol. Each item comes with a literal instruction, a knowledge-based (indirect) instruction for the same target, and matched alternative versions of both that refer to a second object. The literal contrast measures how much changing the named object changes which object is lifted, and the knowledge contrast measures the same effect for the indirect description. Their pooled ratio over a fixed set of items that a language probe answers correctly and that the literal-adapted policy controls is the *contrastive knowledge utilization* (KU).

To reduce the gap, we propose *Referential Self-Distillation* (RSD). A language probe of the frozen backbone translates the indirect description into an object name, and an EMA teacher conditions on the resulting literal instruction. The student receives the indirect instruction and matches 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.

teacher's flow-matching velocity field on states it visits itself, while a literal behavior-cloning term and a contrastive margin keep it sensitive to the instruction. This adapts context distillation (Snell et al., 2022) to a flow-matching policy (Lipman et al., 2023).

Additional teacher-labeled data alone can improve indirect execution, and KU can rise when literal control falls. The DAgger control therefore matches RSD's update budget, collection protocol, gate, schedule, weighting, and literal anchor; actual visited states can differ because each policy controls its own episodes. DAgger uses sampled teacher actions and no alternative-name margin. Under teacher self-consistency, sample cloning and RSD have equal expected gradients for a fixed teacher and state distribution, with higher conditional covariance for cloning. The no-margin ablation is the closer empirical test of the velocity target, though it too can visit different states.

Our contributions are:

- We define KnowAct and contrastive knowledge utilization, which condition action measurement on reported knowledge and literal control while canceling instruction-independent additive selection biases (Section 3).
- We introduce RSD for flow-matching VLAs and show that, under a self-consistent teacher, its distillation term is a Rao-Blackwellized form of on-policy sample cloning (Section 4).
- We evaluate two VLA architectures with budget-matched controls, held-out knowledge types and objects, fictional-fact injection, readout probes, and activation patching. RSD has higher pooled KU than DAgger on both architectures and positive paired KU differences on the evaluated GR00T held-out splits, while intervention evidence links decodable referent states to action selection (Sections 5 and 6).

2 Related Work

Knowledge in Robot Policies. VLA models connect vision-language pretraining to control through different interfaces: RT-2 and OpenVLA emit action tokens from the fine-tuned backbone (Brohan et al., 2023; Kim et al., 2024), π_0 and $\pi_{0.5}$ attach a flow-matching action expert (Black et al., 2025; Physical Intelligence et al., 2025), and GR00T N1 reads an intermediate layer with a diffusion-transformer head (Bjorck et al., 2025). Act2Answer turns knowledge questions into action-based answer selection (Kachaev et al., 2026) and RoboSemanticBench diagnoses semantic grounding in action prediction (Yu et al., 2026); KnowAct conditions its primary score on item-level knowledge and on literal control, so its denominators differ from both.

Instruction Sensitivity. LIBERO-Para (Kim et al., 2026), built on LIBERO (Liu et al., 2023), separates paraphrases of the action from changes to the object reference. LangForce counters a shortcut in which the policy ignores language by adding a conditional information objective (Lian et al., 2026), and GSR re-binds instruction semantics to grounded representations (Yin & Zhang, 2026). These methods treat insensitivity to the instruction as a failure mode to be trained away; KnowAct instead measures literal instruction sensitivity and uses it as the denominator of KU.

Explicit Routes and Distillation. Hierarchical systems first reason or plan in language and then act (Ahn et al., 2022; Zawalski et al., 2024), and VLA Grounder optimizes the language condition of a frozen policy at test time (Shodiev et al., 2026); our explicit-translation reference is a minimal instance of that family. Context distillation trains a model to reproduce, without the context, the outputs it produces with extra information in context (Snell et al., 2022); it extends classical distillation (Hinton et al., 2015), and self-distillation can use trajectories generated by the student (Shenfeld et al., 2026). RSD applies context distillation to the velocity field of a flow-matching policy (Lipman et al., 2023) on learner-visited states, following the state-collection argument of DAgger (Ross et al., 2011).

Knowing versus Using. Language models that store a fact can fail to classify, compare, or reverse it (Allen-Zhu & Li, 2023; Berglund et al., 2024), and latent two-hop reasoning requires both recall and composition (Balesni et al., 2024). Probe accuracy depends on probe capacity and must be read against control tasks (Hewitt & Liang, 2019; Belinkov, 2022), so KnowAct uses language answers only to select and stratify items for behavioral measurement.

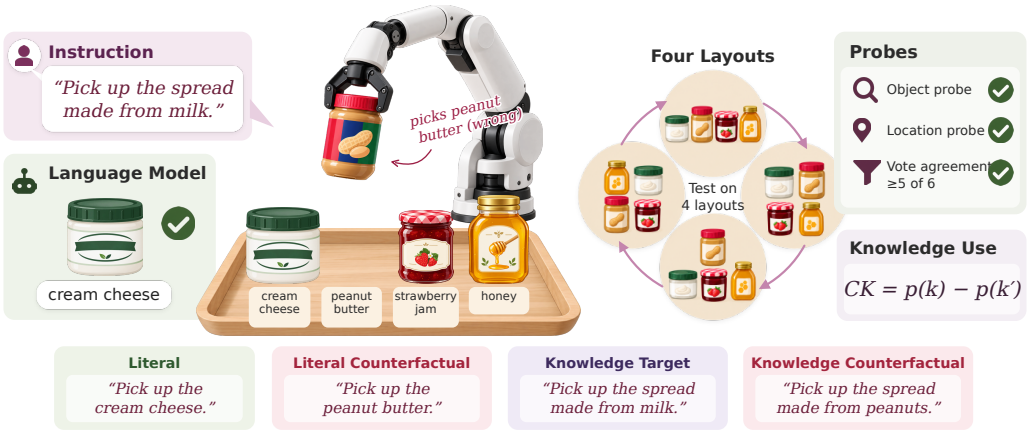


Figure 1: KnowAct measurement overview. The language probe can identify the object named by an indirect instruction even when the policy selects another object. Each item has four cyclic layouts and four matched instructions: literal target and alternative names, and indirect target and alternative references. Object and location probes assess reported knowledge; the knowledge contrast compares target-selection probabilities under the two indirect instructions. The fixed screened set S, literal contrast, and KU estimator are defined in the text. Instructions in the diagram omit the shared tray-placement clause.

3 Measuring Knowledge Utilization

Items and Outcomes. A policy maps an observation o_t and an instruction x to an action chunk $a_{t:t+H}$. A KnowAct item (Figure 1) is a tabletop scene with four individually nameable objects and a receptacle. It designates a target and an alternative object and supplies four instructions differing only in the noun phrase: l_i and l_i' name the target and the alternative object, and k_i and k_i' refer to the same two objects through a knowledge predicate, for example "the spread made from milk" and "the spread made from peanuts." Four cyclic layouts place every object in every slot once. The selected object is the first one lifted stably above its support, and an episode without such a lift counts as a failure and stays in the denominator. We write $p_i(x)$ for the probability that the target is selected under x.

Knowledge Probes. The explicit probe V_i^{exp} asks which object matches the indirect description, and the location probe V_i^{lat} asks which slot holds that object. Each scores a closed candidate set, the four registered object names or the four slot labels, by length-normalized answer log-probability under six prompt-and-order combinations, and an answer counts as confident knowledge if it is correct and at least five votes agree (Appendix B). Both probes receive the policy's processed image. The GR00T probe queries its frozen language head above the action-head readout; the $\pi_{0.5}$ probe uses the source VLM and therefore measures source knowledge only.

Contrasts and Utilization. On an item set T, the literal contrast, the knowledge contrast, and the contrastive knowledge utilization are

$$\mathrm{CL}_i = p_i(l_i) - p_i(l_i'), \qquad \mathrm{CK}_i = p_i(k_i) - p_i(k_i'), \qquad \mathrm{KU}(T) = \frac{\sum_{i\in T} \mathrm{CK}_i}{\sum_{i\in T} \mathrm{CL}_i}. \tag{1}$$

When reporting a set-level literal or knowledge contrast, we mean the arithmetic item mean $\overline{\mathrm{CL}}(T) = |T|^{-1}\sum_{i\in T}\mathrm{CL}_i$ or $\overline{\mathrm{CK}}(T) = |T|^{-1}\sum_{i\in T}\mathrm{CK}_i$, so $\mathrm{KU}(T) = \overline{\mathrm{CK}}(T)/\overline{\mathrm{CL}}(T)$. Later shorthand such as "CK on S" refers to this mean, not the sum. KU compares the pooled knowledge-reference sensitivity with pooled literal-reference sensitivity. When every $\mathrm{CL}_i > 0$, it is a CL_i-weighted average of item ratios; otherwise that weighted-average interpretation does not apply, although the ratio of sums remains defined when its denominator is positive. The pooled ratio avoids unstable per-item ratios. Both contrasts are differences, so any preference that shifts target probability equally under all four instructions cancels from CL_i, from CK_i, and hence from KU: slot, salience, and

object preferences of that form do not affect the score, and the balanced layouts also deny any object a privileged slot. Preferences that interact with the wording do not cancel, and neither do preferences additive in the logit (Appendix C.3).

Fixed Conditioning Set. The primary set S is fixed once per model from a literal-adapted initialization S0: an item enters S if the explicit probe is confidently correct and $\mathrm{CL}_i^{\mathrm{S0}} \geq 0.6$ in a separate S0 screening evaluation. The stated screening rule pools the three seeded S0 runs before applying the threshold (Appendix C). All later methods are evaluated on that same S, each contributing its own literal denominator, and because a method can raise KU by degrading literal control, CL and CK accompany every KU value. Evaluating the literal instruction in place of the indirect one gives $\mathrm{KU} = 1$ by construction, so one is a reference level rather than an upper bound.

Location Strata. The location probe partitions S into $S_+ = \{i \in S : V_i^{\mathrm{lat}} = 1\}$ and $S_- = S \setminus S_+$.

Proposition 1 (Stratum Decomposition)**.** *If both stratum denominators* $\sum_{i \in S_+} \mathrm{CL}_i$ *and* $\sum_{i \in S_-} \mathrm{CL}_i$ *are positive, then* $\mathrm{KU}(S) = w\,\mathrm{KU}(S_+) + (1-w)\,\mathrm{KU}(S_-)$ *with* $w = \sum_{i \in S_+} \mathrm{CL}_i / \sum_{i \in S} \mathrm{CL}_i \in (0, 1)$.

The weight w is recomputed for every method and can differ from the fraction $|S_+|/|S|$ of correct location answers (Appendix C.3). A low $\mathrm{KU}(S_+)$ would show that knowledge the language head can localize still fails to guide selection, but because a language prompt and an action head read the representation differently, it would not locate where the information is lost.

Secondary Quantities. Belief-action agreement (BAA) is the probability that executing k_i selects the object named by the explicit probe, so $\mathrm{BAA}_i = p_i(k_i)$ on S. Adding a constant to both $p_i(k_i)$ and $p_i(k_i')$ raises mean BAA and leaves KU unchanged: BAA credits a policy drawn toward the target under any indirect instruction, whereas KU credits only the selectivity separating the two references. Because conditioning limits the comparison to items the model knows and controls, every method is additionally scored on all 480 items; that score mixes knowledge-positive and knowledge-negative items, so a gain there may reflect newly learned description-to-object associations rather than better use of reported knowledge.

4 Referential Self-Distillation

RSD (Figure 2) continues training from S0. It teaches the direct route, which receives the indirect instruction, to act as the explicit route acts after translating that instruction into an object name, with the backbone frozen and only the action-side modules updated.

Translation and Teacher. The language component of the frozen backbone scores the registered object names under five prompts and accepts a name when at least four select it by a sufficient margin (Appendix B). The accepted name forms a literal instruction $\hat{l}$, a different object supplies $\hat{l}_{\mathrm{neg}}$, and because the gate never reads the evaluation label a confidently wrong translation can pass it. The teacher is an exponential moving average of the student's trainable parameters and shares the frozen backbone; during collection it controls an episode with a probability that decays to zero over the first 30% of continuation updates. Each accepted episode stores its visited states with k, $\hat{l}$, $\hat{l}_{\mathrm{neg}}$, and a confidence weight $w(o, k)$ (Appendix D).

Velocity Objective. We use a canonical flow time s with noise at $s = 0$ and actions at $s = 1$; model-specific adapters convert native time and velocity conventions and call the native training-time sampler for s. A teacher chunk $a_T \sim \pi_\phi(\cdot \mid o, \hat{l})$ and independent Gaussian noise ϵ define the interpolant $x_s = (1-s)\epsilon + s\,a_T$, and $D_M(u, v)$ is the root-mean-square distance over the action horizon and the valid, unpadded coordinates. With the student velocity $v_S = v_\theta(x_s, s \mid o, k)$ and the teacher velocities $v_T = v_\phi(x_s, s \mid o, \hat{l})$ and $v_N = v_\phi(x_s, s \mid o, \hat{l}_{\mathrm{neg}})$, the loss is

$$\begin{aligned}\mathcal{L} = \mathcal{L}_{\mathrm{BC}} &+ \lambda_d\, \mathbb{E}\big[w(o,k)\, D_M(v_S, \mathrm{sg}[v_T])^2\big] \\ &+ \lambda_n\, \mathbb{E}\big[\max\big(0, \gamma D_M(\mathrm{sg}[v_T], \mathrm{sg}[v_N]) - D_M(v_S, \mathrm{sg}[v_N])\big)\big],\end{aligned} \tag{2}$$

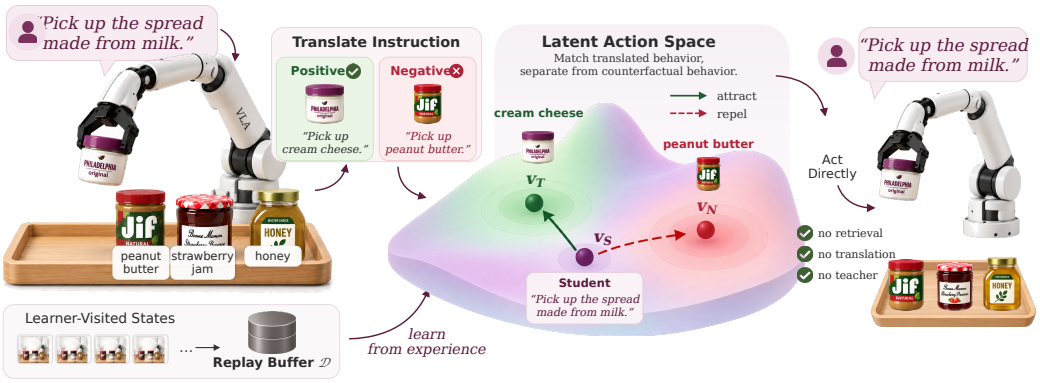


Figure 2: Referential Self-Distillation. A gated translation supplies positive and alternative literal names for learner-visited states. In the latent action space, the student conditioned on the indirect instruction matches the teacher velocity for the positive name and maintains a margin from the alternative velocity; literal behavior cloning anchors the update. At deployment the student acts directly on the indirect instruction without translation or teacher. Instruction strings omit the shared tray-placement clause.

where sg stops gradients, $\lambda_d = 1$, $\lambda_n = 0.1$, and $\gamma = 0.5$, and $\mathcal{L}_{\mathrm{BC}}$ is flow-matching behavior cloning on literal demonstrations, which fill half of every batch. The margin penalizes a student velocity within $\gamma D_M(v_T, v_N)$ of the alternative teacher velocity, and its scale shrinks where the teacher prescribes similar early motion for both objects. A velocity far from both teachers also satisfies the margin, so the margin alone cannot prevent an instruction-independent solution; the distillation term and the literal anchor supply that constraint, and CL measures whether instruction sensitivity survives (Appendix D).

Relation to Sample Cloning. DAgger-style sample cloning replaces the distillation term with flow-matching regression onto the teacher's sampled chunk, $\mathbb{E}[w\, D_M(v_S, a_T - \epsilon)^2]$, on the same learner-visited states. Call the teacher self-consistent at the stored states if $v_\phi(x, s \mid o, \hat{l}) = \mathbb{E}[a_T - \epsilon \mid x_s = x, s, o, \hat{l}]$ under the interpolant above.

Proposition 2 (Rao-Blackwellized Cloning)**.** *For a fixed, self-consistent teacher and state distribution, with stored confidence weights independent of update-time action samples, the two terms differ by a constant independent of θ, so their expected gradients are equal. The cloning gradient equals the RSD gradient plus a conditionally zero-mean term, so its conditional covariance is at least as large.*

RSD replaces $a_T - \epsilon$ by its conditional expectation given $(x_s, s, o, \hat{l})$ (Appendix C.3). Under the proposition's conditions the objectives share their minimizers and RSD reduces gradient variance. A coarse teacher solver or approximate fit can break self-consistency, so observed differences can reflect finite optimization, the margin, inconsistency, or visited states. Removing the margin makes the RSD-DAgger comparison more informative about the velocity target, but the trained policies can still visit different states and learn different teachers.

Inference. At test time the RSD student receives the observation and the indirect instruction and runs the same action-head computation as S0, without translator or teacher. The explicit-translation reference calls the translator once per episode and executes S0 with the resulting literal instruction, falling back to the indirect one when the gate rejects. The frozen backbone cannot change during continuation training, but the action head can still learn a new mapping from features a language probe fails to decode, which the held-out splits and the injection test constrain.

5 EXPERIMENTAL SETUP

We evaluate KnowAct and RSD on GR00T N1.7 and $\pi_{0.5}$ in RoboTwin 2.0. The benchmark, model interfaces, training run matrix, and estimators are specified in Appendices A, C and D.

Models and Interfaces The GR00T policy starts from `nvidia/GR00T-N1.7-3B` and uses backbone layer 12 instead of the public layer 16; S0 trains its action-side interface for that readout. The $\pi_{0.5}$ policy starts from `lerobot/pi05_base` (NVIDIA, 2026b; Hugging Face LeRobot, 2026; Physical Intelligence et al., 2025). Both backbones stay frozen during our training. GR00T's knowledge probe uses the same backbone with its language head. Because robot pretraining changed the $\pi_{0.5}$ backbone, its probe uses the source `google/paligemma-3b-pt-224` (Beyer et al., 2024); this arm provides an architecture replication under a source-VLM knowledge reference.

Benchmark Splits All items are rendered in RoboTwin 2.0 (Chen et al., 2025) with an ALOHA-style bimanual embodiment. The left arm always acts, and all four objects are within its workspace. Of 480 items, TE has 200 items with training knowledge types but unseen object combinations; HT has 160 with held-out origin, OCR, and reasoning types; HO has 120 with training types but object identities absent from training. Each item uses four instructions, four layouts, two repetitions, and three training seeds: 24 rollouts per instruction and 46,080 episodes per model-method condition (Appendix A).

Training Conditions Each architecture has S0 runs with seeds 0, 1, and 2, trained for 20,000 updates on 10,240 literal demonstrations. Each continuation starts from its seed-matched S0 checkpoint and trains for 8,000 updates at batch size 32. Literal continuation, paraphrase BC, Relabel-BC, DAgger, RSD, offline RSD, and RSD without margin vary indirect supervision and collection as specified in Table 1. GR00T uses the full set; $\pi_{0.5}$ uses literal continuation, DAgger, and RSD. The main run matrix contains six S0 and 30 continuation runs (Appendix D).

Knowledge Injection On 120 separate items, 24 object identities receive five fictional-manufacturer references each. Two shared LoRA adapters (Hu et al., 2022) train on 14,400 VQA examples without action supervision. The lower adapter modifies language layers 1-12, at or below the GR00T readout; the upper modifies layers 13-28 and serves as a structural control. A third condition places the fact in the instruction. Injection passes only when name and location probes both reach 0.90 accuracy. We evaluate all 120 items without S0 screening (Appendix E).

Representation Analyses E1 trains a linear slot probe on indirect-instruction readout states and compares it with a random-label control (Hewitt & Liang, 2019). E3 patches reference-span states with literal target-name states, a whole-instruction control, a generic steering vector, or an alternative name (Meng et al., 2022). Both analyses leave the evaluated policies fixed (Appendix F).

Analysis For each item and instruction, $\hat{p}_i(x)$ averages target selection over 24 rollouts; no lift counts as non-selection. We resample items, pairing methods by item and conditioning on the fixed screening set and three trained runs. The primary estimand is pooled GR00T RSD minus DAgger $\mathrm{KU}(S)$, interpreted with CK and CL guardrails. Split, stratum, and run comparisons without component intervals are descriptive. Explicit translation executes S0 whereas literal continuation has 8,000 more updates, so their headline comparison is operational rather than a same-checkpoint causal contrast. Appendix I.1 reports a matched checkpoint control.

6 EXPERIMENTAL RESULTS

We first quantify the gap between identifying a referent and selecting it during action. We then compare supervision methods, evaluation splits, training components, and representation interventions. Literal continuation and explicit translation serve as diagnostic references; the primary indirect-training comparison is RSD against DAgger.

The primary GR00T RSD-DAgger interval uses the paired item bootstrap in Appendix C.5; the 157-item effect summary appears in Appendix H.5.

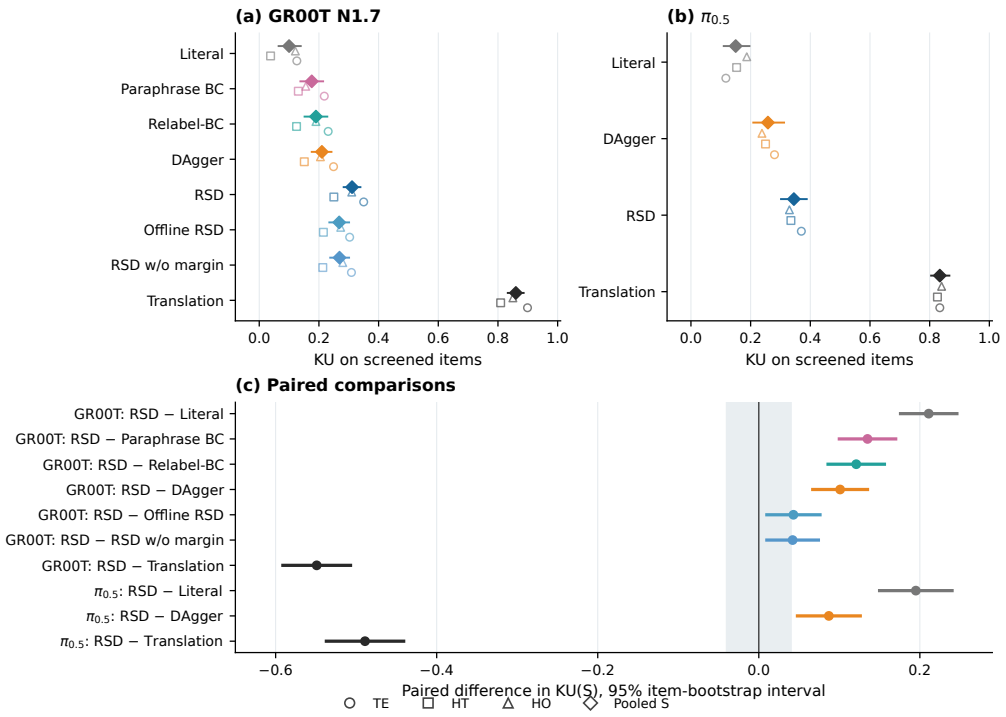


Figure 3: (a,b) KU by condition and split for GR00T N1.7 with $|S| = 157$ and $\pi_{0.5}$ with $|S| = 129$. Markers distinguish TE, HT, HO, and the pooled set. (c) RSD minus each control in pooled $\mathrm{KU}(S)$; intervals show paired item-bootstrap uncertainty for the displayed differences. Literal continuation and translation are diagnostic references. The shaded ± 0.04 band is a visual guide, not an inferential threshold.

Knowledge-Action Gap For GR00T N1.7, the operational explicit-translation reference, which executes S0, reaches $\mathrm{KU}(S) = 0.860$, compared with 0.100 for post-continuation literal control (Figure 3). Their paired difference is 0.760 $[0.712, 0.805]$; the CK difference is 0.558 $[0.523, 0.592]$ and the CL difference is -0.015 $[-0.035, 0.004]$. Similar CL rules out a large aggregate literal-control change, but the different checkpoints make this operational rather than same-checkpoint causal evidence.

The operational translation-literal gap occurs in all three GR00T splits. Explicit translation attains KU values of 0.899, 0.809, and 0.850 on TE, HT, and HO, while literal continuation attains 0.126, 0.038, and 0.121. These comparisons concern the fixed knowledge-positive, literal-control-positive screen S; they do not estimate performance on all 480 benchmark items.

6.1 Supervision Comparisons

Under the shared continuation schedule, GR00T point estimates rise from 0.100 for literal continuation to 0.176 for paraphrase BC, 0.190 for Relabel-BC, 0.210 for DAgger, and 0.311 for RSD. These are distinct supervision recipes rather than cumulative additions. RSD's paired KU differences are 0.135 $[0.100, 0.170]$, 0.121 $[0.086, 0.156]$, and 0.101 $[0.067, 0.135]$ against paraphrase BC, Relabel-BC, and DAgger. The ± 0.04 visual band is not used for inference.

Relative to literal continuation, RSD improves pooled KU by 0.211 $[0.176, 0.246]$ in the diagnostic comparison. Its knowledge contrast rises from 0.075 to 0.230, while literal contrast changes from 0.751 to 0.740 (Figure 4a). Under the target indirect instruction, RSD selects the target in 38.0% of rollouts, compared with 30.2% for DAgger (Figure 4b). Against the budget- and collection-protocol-matched DAgger control, knowledge contrast increases by 0.077 $[0.049, 0.105]$ while the literal-contrast difference is 0.011 $[-0.012, 0.033]$. The gain is therefore not produced by a falling pooled literal denominator in this comparison.

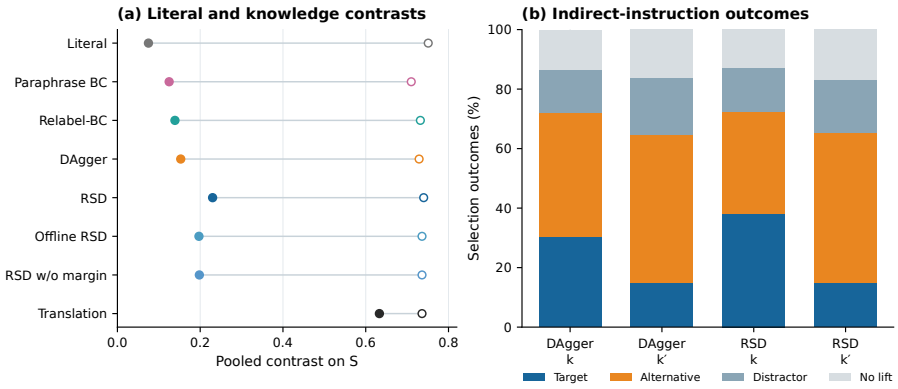


Figure 4: (a) Pooled literal (open) and knowledge (filled) contrasts on the GR00T conditioning set; paired markers share a method row. (b) Observed selection outcomes under the two indirect instructions for DAgger and RSD, each based on 3,768 rollouts.

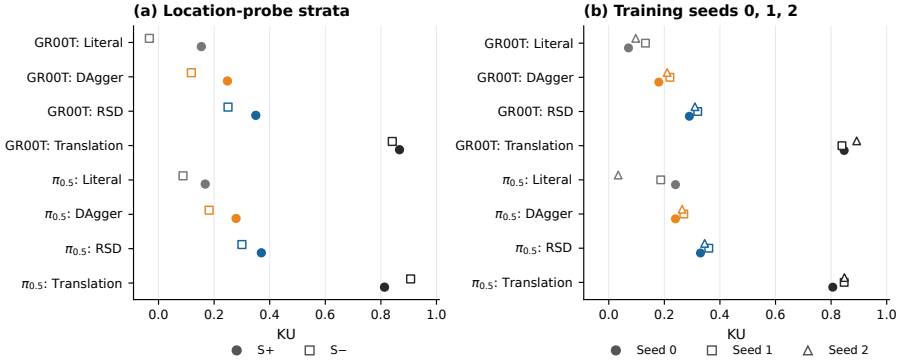


Figure 5: KU by location-probe stratum (a) and training seed (b). Filled circles and open squares denote S_+ and S_-; the seed panel shows independently trained runs 0, 1, and 2. Points are descriptive; stratum and retraining intervals are not shown.

Transfer across Splits and Architectures RSD exceeds DAgger in KU on TE, HT, and HO by 0.101 $[0.045, 0.157]$, 0.099 $[0.031, 0.167]$, and 0.105 $[0.035, 0.175]$. Each split-specific paired item interval is above zero. The comparison supports consistency across the evaluated held-out splits, while leaving transfer to arbitrary new knowledge types or object categories untested. On $\pi_{0.5}$, RSD, DAgger, and literal continuation reach 0.345, 0.258, and 0.150. RSD exceeds DAgger in KU by 0.087 $[0.048, 0.126]$ and literal continuation by 0.195 $[0.150, 0.240]$, while explicit translation reaches 0.834. This is a KU-only replication under a source-VLM reference because component guardrails are unavailable.

Stratification and Run Variation The point estimates favor RSD in both strata (Figure 5a): RSD versus DAgger is 0.350 versus 0.248 on S_+ and 0.250 versus 0.118 on S_- for GR00T N1.7, and 0.370 versus 0.279 and 0.300 versus 0.182 for $\pi_{0.5}$. These comparisons are descriptive because stratum-specific intervals are unavailable. RSD also exceeds DAgger for each of the three training seeds on both backbones, with RSD KU spanning 0.290-0.320 on GR00T and 0.330-0.360 on $\pi_{0.5}$. The unavailable retraining intervals limit this comparison to the evaluated runs.

Collection and Margin Ablations Online RSD exceeds offline RSD by 0.043 $[0.010, 0.076]$ in KU. Full RSD exceeds the no-margin variant by 0.042 $[0.010, 0.074]$, with pooled CL 0.740 versus 0.736. Both item intervals are positive, but neither quantifies retraining variation or isolates the velocity target from other differences between trained policies.

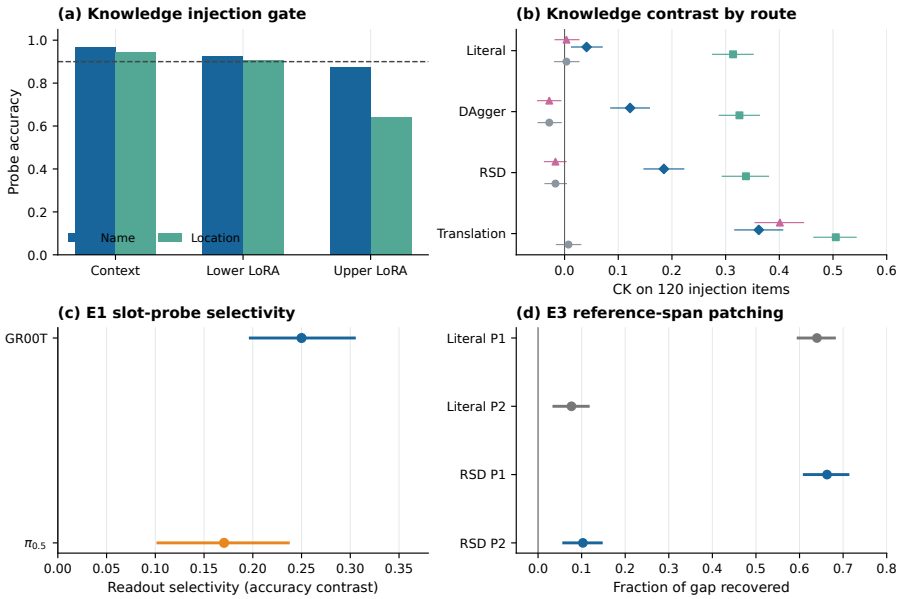


Figure 6: Injection and representation results.

Representation Evidence Context and lower-adapter injection pass both 0.90 probes; the upper adapter fails (Figure 6). Under the lower adapter, RSD reaches CK $= 0.185$ $[0.148, 0.222]$ versus DAgger's 0.122 $[0.086, 0.158]$, a paired difference of 0.063 $[0.031, 0.095]$ from one adapter seed. Readout selectivity is 0.250 $[0.197, 0.304]$ for GR00T N1.7 and 0.171 $[0.103, 0.236]$ for $\pi_{0.5}$. The corresponding RSD decoding-contrast correlations are 0.33 $[0.18, 0.46]$ and 0.30 $[0.13, 0.45]$; they measure association, not feature use. Reference-span patching recovers 0.640 $[0.597, 0.680]$ and 0.663 $[0.611, 0.711]$ of the literal and RSD gaps, versus 0.077 and 0.103 for generic steering; alternative-name patches drive KU to -0.653 and -0.590.

Additional Control Experiments At S0, direct and translated execution have CK 0.081 and 0.633 with common CL 0.736. The four factorial KU values are 0.210, 0.233, 0.269, and 0.311 for sampled actions without and with margin, then teacher velocity without and with margin. The no-margin velocity gap is 0.059, but online training also changes teachers and visited states (Appendices I.2 and I.7). Among 157 GR00T items, paired CK changes are positive on 110, negative on 35, and zero on 12; median $\Delta\mathrm{CK}_i = 0.0833$ and the paired ΔKU interval is $[0.067, 0.135]$ (Appendix H.5). Further controls appear in Appendix I.

7 Conclusion

KnowAct measures a gap between reported knowledge and action in simulated manipulation. Explicit translation exceeds literal continuation by 0.760 in KU across different checkpoints. RSD exceeds matched DAgger by 0.101 on GR00T N1.7 and 0.087 on $\pi_{0.5}$, with positive paired intervals and GR00T split differences. Item effects remain heterogeneous, and the controls do not isolate a single mechanism.

Limitations. Results come from simulation rather than physical deployment. Item-bootstrap intervals condition on 157 screened items and three trained runs, leaving retraining variation unmeasured. The main injection result uses one adapter seed; $\pi_{0.5}$ uses a source-VLM probe; and translation and literal references use different checkpoints. Effects may vary with screening, object composition, and readout choice.

Ethics Statement

The protocol uses simulated tabletop scenes, everyday object properties, and fictional manufacturers. Its references require no stereotypes about people or protected groups, and it involves no human subjects or personal data. The injection procedure writes new associations into model weights. We restrict it to fictional facts about simulated objects and do not deploy injected facts on physical robots. Any implementation must respect the licenses and access conditions of the models and assets it uses.

Reproducibility Statement

Appendix A specifies benchmark construction, splits, leakage controls, layouts, and selection scoring. Appendix B gives the probe and translation prompts and scoring rules. Appendix C defines the estimators, decision rules, and paired item bootstrap. Appendix D records model revisions, training settings, the run matrix, collection schedule, and flow-time conventions. Appendices E and F describe injection, probing, and patching; Appendices G to I report the additional results and controls. Each figure and table identifies its conditioning set and statistical unit where relevant. The primary item bootstrap conditions on the three trained runs and therefore does not estimate full retraining variability.

AI Use Statement

An AI assistant was used to retrieve literature and public model metadata, to suggest the organization of the paper, to help specify parts of the evaluation design, including the decision rules and the probe and translation templates, to draft the propositions and their proofs, to prepare figures, and to draft and edit the text, including the presentation and interpretation of reported results. The authors verified the experimental results and take full responsibility for all content, including the proofs and the evaluation design.

References

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, et al. Do As I Can, Not As I Say: Grounding Language in Robotic Affordances. In *Proceedings of the 6th Conference on Robot Learning*, 2022. URL https://arxiv.org/abs/2204.01691.

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of Language Models: Part 3.2, Knowledge Manipulation. *arXiv preprint arXiv:2309.14402*, 2023. URL https://arxiv.org/abs/2309.14402.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2.5-VL Technical Report. *arXiv preprint arXiv:2502.13923*, 2025. URL https://arxiv.org/abs/2502.13923.

Mikita Balesni, Tomek Korbak, and Owain Evans. Lessons from Studying Two-Hop Latent Reasoning. *arXiv preprint arXiv:2411.16353*, 2024. URL https://arxiv.org/abs/2411.16353.

Yonatan Belinkov. Probing Classifiers: Promises, Shortcomings, and Advances. *Computational Linguistics*, 48(1):207–219, 2022.

Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. The Reversal Curse: LLMs Trained on "A is B" Fail to Learn "B is A". In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2309.12288.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, et al. PaliGemma: A versatile 3B VLM for transfer. *arXiv preprint arXiv:2407.07726*, 2024. URL https://arxiv.org/abs/2407.07726.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, et al. GR00T N1: An Open Foundation Model for Generalist Humanoid Robots. *arXiv preprint arXiv:2503.14734*, 2025. URL https://arxiv.org/abs/2503.14734.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, et al. π_0: A Vision-Language-Action Flow Model for General Robot Control. In *Robotics: Science and Systems*, 2025. URL https://arxiv.org/abs/2410.24164.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, et al. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183, 2023. URL https://arxiv.org/abs/2307.15818.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, et al. RoboTwin 2.0: A Scalable Data Generator and Benchmark with Strong Domain Randomization for Robust Bimanual Robotic Manipulation. *arXiv preprint arXiv:2506.18088*, 2025. URL https://arxiv.org/abs/2506.18088.

John Hewitt and Percy Liang. Designing and Interpreting Probes with Control Tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 2733–2743, 2019.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531*, 2015. URL https://arxiv.org/abs/1503.02531.

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. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2106.09685.

Hugging Face LeRobot. pi05_base Model Card. https://huggingface.co/lerobot/pi05_base, 2026. Accessed September 23, 2026.

Nikita Kachaev, Andrey Moskalenko, Matvey Skripkin, Nikita Kurlaev, Daria Pugacheva, Albina Burlova, et al. Does VLA Even Know the Basics? Measuring Commonsense and World Knowledge Retention in Vision-Language-Action Models. *arXiv preprint arXiv:2606.19297*, 2026. URL https://arxiv.org/abs/2606.19297.

Chanyoung Kim, Minwoo Kim, Minseok Kang, Hyunwoo Kim, and Dahuin Jung. LIBERO-Para: A Diagnostic Benchmark and Metrics for Paraphrase Robustness in VLA Models. *arXiv preprint arXiv:2603.28301*, 2026. URL https://arxiv.org/abs/2603.28301.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, et al. OpenVLA: An Open-Source Vision-Language-Action Model. In *Proceedings of the 8th Conference on Robot Learning*, 2024. URL https://arxiv.org/abs/2406.09246.

Shijie Lian, Bin Yu, Xiaopeng Lin, Laurence T. Yang, Zhaolong Shen, Changti Wu, et al. LangForce: Bayesian Decomposition of Vision Language Action Models via Latent Action Queries. *arXiv preprint arXiv:2601.15197*, 2026. URL https://arxiv.org/abs/2601.15197.

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

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, Datasets and Benchmarks Track*, 2023. URL https://arxiv.org/abs/2306.03310.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and Editing Factual Associations in GPT. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2202.05262.

NVIDIA. Cosmos-Reason2-2B Model Card. https://huggingface.co/nvidia/Cosmos-Reason2-2B, 2026a. Accessed September 23, 2026.

NVIDIA. GR00T-N1.7-3B Model Card. https://huggingface.co/nvidia/GR00T-N1.7-3B, 2026b. Accessed September 23, 2026.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, et al. $\pi_{0.5}$: a Vision-Language-Action Model with Open-World Generalization. *arXiv preprint arXiv:2504.16054*, 2025. URL https://arxiv.org/abs/2504.16054.

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15, pp. 627–635. PMLR, 2011. URL https://proceedings.mlr.press/v15/ross11a.html.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-Distillation Enables Continual Learning. *arXiv preprint arXiv:2601.19897*, 2026. URL https://arxiv.org/abs/2601.19897.

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.

Charlie Snell, Dan Klein, and Ruiqi Zhong. Learning by Distilling Context. *arXiv preprint arXiv:2209.15189*, 2022. URL https://arxiv.org/abs/2209.15189.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, et al. InternVL3.5: Advancing Open-Source Multimodal Models in Versatility, Reasoning, and Efficiency. *arXiv preprint arXiv:2508.18265*, 2025. URL https://arxiv.org/abs/2508.18265.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, et al. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Zhaokai Yin and Zhipeng Zhang. Grounded Semantic Re-Binding for Robust Instruction Generalization in Vision-Language-Action Models. *arXiv preprint arXiv:2608.02497*, 2026. URL https://arxiv.org/abs/2608.02497.

Bin Yu, Yao Zhang, Haishan Liu, Shijie Lian, Yuliang Wei, Xiaopeng Lin, et al. RoboSemanticBench: Diagnosing Semantic Grounding in Action Prediction for VLA Models. *arXiv preprint arXiv:2606.02277*, 2026. URL https://arxiv.org/abs/2606.02277.

Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic Control via Embodied Chain-of-Thought Reasoning. In *Proceedings of the 8th Conference on Robot Learning*, 2024. URL https://arxiv.org/abs/2407.08693.

A Benchmark Specification

A.1 Items and References

An item consists of four registered object identities, a target, an alternative object, a receptacle, a knowledge predicate, and four instruction strings; the remaining two objects are distractors. Each object has a unique literal name within the scene, and when two objects share a category name, visually grounded qualifiers are added and used identically in all conditions. Every instruction follows the template "Pick up the [object reference] and place it in the tray," and only the reference changes: the two literal references name the target and the alternative object, and the two indirect references identify the same two objects through the same knowledge predicate.

Each knowledge type constrains the references it admits: a functional reference requires that no other object in the scene plausibly serves the same function, a composition predicate concerns an established property of the depicted object rather than an unseen brand variant, OCR references require text legible in the processed image, origin references are tied to a specific documented product, and relative physical properties require a comparison that is well defined within the scene. These constraints keep an ambiguous reference from being scored as a failure to use knowledge.

Items are assembled in three stages: predicates are filtered symbolically so that exactly one object in the scene satisfies each reference, Qwen3-32B (Yang et al., 2025) proposes candidate indirect descriptions, and InternVL3.5-38B (Wang et al., 2025) and Qwen2.5-VL-32B (Bai et al., 2025) each check the rendered scene. An item is kept only if both visual checks agree with the symbolic filter. These models prepare data only; they are distinct from the policy backbones and supply no knowledge to RSD.

Scene composition is constrained by referential uniqueness and control feasibility, and the four objects need not share a category, since a narrow shared category would conflict with unique literal names and with predicates such as function. Target and alternative descriptions have comparable length and syntax, and counterbalancing both identities across slots removes a spatial shortcut. Lexical and predicate asymmetries remain possible sources of item heterogeneity, which the per-type and per-layout summaries expose.

A.2 Example Items

The following examples show one item per knowledge type. Each lists the four objects, with the target first and the alternative object second, followed by the two indirect references.

Composition (Training Type). Cream cheese, peanut butter, strawberry jam, honey. "The spread made from milk" and "the spread made from peanuts."

Function (Training Type). Thermometer, measuring tape, stapler, sponge. "The object used for measuring temperature" and "the object used for measuring length."

Physical Attribute (Training Type). Glass cup, rubber ball, wooden block, paper bag. "The object that shatters when dropped" and "the object that bounces when dropped."

Origin (Held-Out Type). Parmesan wedge, Camembert box, cracker box, apple. "The cheese that originates from Parma" and "the cheese that originates from Normandy."

OCR (Held-Out Type). Box labeled SALT, box labeled SUGAR, mug, spoon. "The box whose label reads salt" and "the box whose label reads sugar."

Simple Reasoning (Held-Out Type). Apple, grape, mug, spoon. "The larger of the two fruits" and "the smaller of the two fruits."

The literal references are the object names, for example "the cream cheese" and "the peanut butter." HO items use the three training types with object identities that never occur in S0 demonstrations or in any continuation data.

A.3 Splits and Leakage Controls

TE contains 67 composition, 67 function, and 66 physical-attribute items; HT contains 53 origin, 53 OCR, and 54 simple-reasoning items; HO contains 40 items of each training type. The indirect

training pool contains 120 items of each training type, and the injection set contains 120 further items from 24 objects with five references each. TE uses the training knowledge types with object combinations and indirect descriptions disjoint from training, HT uses knowledge types absent from every indirect training instruction, and HO uses the training types with object identities excluded from all training data, so its score need not lie between those of TE and HT.

The indirect training pool draws on 256 training layouts, where a layout is identified by object identities and geometry rather than by the assignment of objects to nominal slots. TE excludes training object tuples even when their slot assignment differs, and HO excludes individual object identities. Fixed template tokens are removed before the four-gram overlap check between training and evaluation descriptions, no overlapping four-gram is allowed, and descriptions shorter than four tokens are checked for exact duplication instead.

A.4 Layouts and Rollouts

Each evaluation item uses four cyclic assignments of objects to slots: for object index j and layout index b, the object occupies slot $(j + b) \bmod 4$. Each layout is rolled out twice, giving eight rollouts per instruction and seed. Layout and repetition identifiers, including the environment initialization, are shared across methods, while action sampling keeps its own noise seed; object identities define the selection outcome and slot numbers only balance positions. One model-method condition requires $480 \times 4 \times 4 \times 2 \times 3 = 46{,}080$ selection episodes. Items are never moved to another split after evaluation, and the sizes of the knowledge-conditioned sets follow from the probe and S0 rules. The injection set is disjoint from TE, HT, and HO, and all 120 injection items are reported without filtering on probe outcomes.

A.5 Observations and Selection

The embodiment is an ALOHA-style bimanual platform in RoboTwin 2.0. The left arm acts in every episode and the right arm holds a fixed home pose, all four objects lie in the left workspace, and this assignment does not depend on the target and is the same for demonstrations and evaluation. The observation contains a base camera, two wrist cameras, and 14 joint and gripper coordinates; the action interface uses 14 joint and gripper targets, with the unused dimensions of each model's padded representation masked. Control runs at 20 Hz with a 400-tick (20 s) timeout, and the policy replans after executing eight actions of each chunk.

The selected object is the first object whose height exceeds its initial support height by 5 cm for ten consecutive control ticks (0.5 s). Contact alone does not count, because transient contact with a distractor need not reflect the intended choice, and placement in the tray is a separate outcome that also depends on transport and release. If two objects first satisfy the lift criterion on the same tick, the episode is scored as an ambiguous selection and counted as a failure. An episode ends at the first unambiguous stable lift or at timeout; there are no retries, human corrections, or resets chosen after an outcome is seen, and every failure is kept.

Stable lift combines target selection with the ability to establish a grasp, so a policy with correct grounding and poor gripper control would show a low knowledge contrast. S0 screening removes this confound on the selected items at initialization, but continuation training could change control ability, so literal performance is measured again for every continuation condition.

B Probe and Translation Prompts

B.1 Scoring

All probes and the translator score a closed candidate set and never generate free text. For a prompt q, image o, and candidate answer $y = (y_1, \ldots, y_m)$ in the model's tokenization, the score is the length-normalized log-probability

$$\sigma(y \mid q, o) = \frac{1}{m} \sum_{t=1}^{m} \log P(y_t \mid q, o, y_{<t}). \tag{3}$$

The vote of a prompt is the candidate with the highest score, computed in a single teacher-forced forward pass per candidate, so no sampling temperature is involved. The image is the processed base-camera view that the policy receives in the canonical layout $b = 0$, so each probe yields one answer per item. GR00T prompts are wrapped in the chat template of its backbone with the image before the text, and PaliGemma prompts are prefixed with `answer en`. Below, `{k}` denotes an indirect reference and `{c1}` to `{c4}` denote candidate names or slot labels.

B.2 Explicit Knowledge Probe

The explicit probe scores the four registered object names in the scene. Each of the three templates is presented with the candidates in slot order and in reverse slot order, which gives six votes.

- `Question: Which object on the table is {k}? Options: {c1}, {c2}, {c3}, {c4}. Answer: the`
- `Consider {k}. Which of these objects is it: {c1}, {c2}, {c3}, or {c4}? It is the`
- `Look at the table. Among {c1}, {c2}, {c3}, and {c4}, the one that is {k} is the`

An item is knowledge-positive if at least five of the six votes name the target. An item with five or more votes for a wrong object has a confident wrong belief and is excluded from S but kept for the unconditional score, and items with fewer than five agreeing votes are treated as unknown. Outside S, belief-action agreement uses the plurality answer as the believed object.

B.3 Location Probe

The location probe scores four slot labels defined relative to the base camera: `front left`, `front right`, `back left`, and `back right`. The three templates are again presented with two label orders, which gives six votes.

- `Question: Where on the table is {k}? Options: {c1}, {c2}, {c3}, {c4}. Answer: the`
- `Consider {k}. In which part of the table is it: {c1}, {c2}, {c3}, or {c4}? It is at the`
- `Look at the table. Among the {c1}, {c2}, {c3}, and {c4} positions, {k} is at the`

$V_i^{\text{lat}} = 1$ requires at least five of the six votes to name the slot that holds the target. The probe answers with a position and does not name the object, so a correct answer requires both the knowledge step and the localization step.

B.4 Translation Prompts

The translator used during RSD collection and by the explicit-translation reference scores the four object names in slot order under five templates.

- `Which object on the table is {k}? Answer: the`
- `The instruction asks for {k}. The requested object is the`
- `Pick up {k}. In other words, pick up the`
- `Objects: {c1}, {c2}, {c3}, {c4}. The one that is {k} is the`
- `Name the object that is {k}: the`

A name is accepted if at least four templates vote for it and the average margin over the five templates between its score and the second-highest exceeds 0.15 nats per token. The set S is selected with the six-vote probe whereas training and the explicit route use this five-template gate, so agreement between the two contributes to the explicit-translation score on S and is reported per architecture as the fraction of items in S whose accepted translation names the target.

B.5 Negative Names and Confidence

The alternative teacher condition $\hat{l}_{\text{neg}}$ uses a name drawn uniformly from the three scene objects other than the accepted name and is resampled for every stored episode, so the margin is not tied to one fixed distractor. The translation confidence entering $w(o, k)$ is the fraction of the five templates voting for the accepted name, which takes the values 0.8 and 1.0 after the gate. On evaluation items the explicit route applies the same gate and executes S0 on the original indirect instruction when the gate rejects, so rejections count against its score.

C Estimation and Proofs

C.1 Selection Probabilities

For item i, seed t, rollout r, and instruction x, let Y_{itrx} be the selected object identity, taking values in $\{0, 1, 2, 3, \varnothing\}$ with object 0 the target, object 1 the alternative object, and $\varnothing$ covering timeout and ambiguous selection. The estimated target probability is

$$\hat{p}_i(x) = \frac{1}{24} \sum_{t=0}^{2} \sum_{r=1}^{8} \mathbb{1}\{Y_{itrx} = 0\}. \tag{4}$$

The literal and knowledge contrasts follow Equation (1). A failure contributes zero to the indicator and stays in the average; dropping failures would estimate selection conditional on a successful lift, a different quantity that can change the ranking of methods. CL also differs from literal success, since a policy that lifts the target under both literal instructions has high target-lift probability and little literal contrast.

For a set T, let $C_T = \sum_{i \in T} \mathrm{CL}_i$ and $K_T = \sum_{i \in T} \mathrm{CK}_i$, so that $\mathrm{KU}(T) = K_T / C_T$. If every $\mathrm{CL}_i > 0$, this ratio is a literal-contrast-weighted mean of the per-item ratios $\mathrm{CK}_i / \mathrm{CL}_i$; with negative item contrasts it is only a ratio of pooled sums, not a convex average. Averaging per-item ratios directly would be unstable near zero and would overweight weak literal contrasts. Finite-sample contrasts lie in $[-1, 1]$, but the ratio can leave $[0, 1]$: a negative numerator means the alternative indirect instruction selects the target more often than the target one, and a value above one means the knowledge contrast exceeds the literal contrast on the same set. Neither is clipped, and if a denominator is not positive, KU is withheld and the unnormalized contrasts are reported.

C.2 Conditioning and Selection Error

S0 determines S before any continuation training from the confident explicit-knowledge indicator and the threshold $\mathrm{CL}_i^{\mathrm{S0}} \geq 0.6$, in a separate screening evaluation of 46,080 episodes per architecture. The screening protocol pools the four layouts, two repetitions, and three S0 training seeds, giving 24 screening rollouts per item and instruction; the screen therefore has one model-specific mask rather than one mask per seed. No continuation condition selects its own evaluation set. The mask is applied unchanged to every method and every bootstrap resample. Because the knowledge probes and initial policies differ between architectures, the primary result describes knowledge-positive items that the initialization controls and within-model paired comparisons carry more information than rankings across models.

The fixed rule avoids method-specific reselection but does not remove selection error: with finitely many S0 rollouts, an item near the control threshold can enter or leave S through sampling noise. The bootstrap conditions on the selected mask and does not resample prompts, so neither the uncertainty of reconstructing S nor prompt uncertainty lies inside the reported intervals.

C.3 Proofs

Additive Bias Invariance. If $p_i(x) = b_i + g_i(x)$, then $\mathrm{CL}_i = g_i(l_i) - g_i(l_i')$ and $\mathrm{CK}_i = g_i(k_i) - g_i(k_i')$; neither involves b_i, and $\mathrm{KU}(T)$ depends on the contrasts only. A preference that depends on the wording does not cancel, and a preference additive in the logit is generally not additive in probability and does not cancel either.

Proof of Proposition 1. Because S_+ and S_- partition S, we have $C_S = C_{S_+} + C_{S_-}$ and $K_S = K_{S_+} + K_{S_-}$. Hence

$$\frac{K_S}{C_S} = \frac{C_{S_+}}{C_S}\frac{K_{S_+}}{C_{S_+}} + \frac{C_{S_-}}{C_S}\frac{K_{S_-}}{C_{S_-}} = w\,\mathrm{KU}(S_+) + (1-w)\,\mathrm{KU}(S_-). \tag{5}$$

When $C_{S_+} > 0$ and $C_{S_-} > 0$, the weight w lies in $(0, 1)$ and the expression is a convex combination. No independence assumption is used. The weights are recomputed for each method, because fixing the item mask does not fix the literal contrasts. □

Belief-Action Agreement. On S the explicit probe names the target, so $\mathrm{BAA}_i = \Pr(Y = 0 \mid k_i) = p_i(k_i)$. Replacing $p_i(k_i)$ and $p_i(k_i')$ by $p_i(k_i) + c$ and $p_i(k_i') + c$ leaves every contrast unchanged, so KU is unchanged while mean BAA over S increases by c. Outside S the believed object can be wrong, and a policy that consistently acts on a wrong answer then has high agreement and low target selection.

Proof of Proposition 2. Write $\|y\|_M^2 = \big(H\sum_d M_d\big)^{-1}\sum_{h,d} M_d\, y_{hd}^2$, so that $D_M(u, v) = \|u - v\|_M$, and let $\langle\cdot,\cdot\rangle_M$ be the associated inner product. Let $z = (x_s, s, o, k, \hat{l})$ and $u = a_T - \epsilon$. Condition on the stored episode and its confidence weight w; the update-time action sample is independent of the samples used to set w. Given z and w, v_S is fixed and self-consistency gives $\mathbb{E}[u \mid z, w] = v_T$. Then

$$\mathbb{E}\big[\|v_S - u\|_M^2 \mid z, w\big] = \|v_S - v_T\|_M^2 + \mathbb{E}\big[\|u - v_T\|_M^2 \mid z, w\big], \tag{6}$$

because the cross term $2\langle v_S - v_T, \mathbb{E}[u \mid z, w] - v_T\rangle_M$ vanishes. Multiplying by w and averaging over the stored states shows that the cloning objective equals the RSD distillation objective plus $\mathbb{E}\big[w\,\|u - v_T\|_M^2\big]$. During a student update the teacher and stored-state distribution are fixed, so this term has zero gradient with respect to θ; no gradient is propagated through the teacher. The per-sample cloning gradient is $2wJ_\theta^\top \bar{M}(v_S - u)$, where J_θ is the Jacobian of v_S with respect to θ and $\bar{M}$ is the diagonal weight matrix of $\|\cdot\|_M^2$. It equals the per-sample RSD gradient $2wJ_\theta^\top \bar{M}(v_S - v_T)$ plus $2wJ_\theta^\top \bar{M}(v_T - u)$, which has conditional mean zero. The conditional covariance of the cloning gradient is therefore the RSD covariance plus a positive semidefinite term. □

Self-consistency holds when v_ϕ equals the marginal velocity of the linear path between Gaussian noise and the teacher's own sample distribution. Exact integration of v_ϕ does not imply it, since several velocity fields can generate the same marginal path, and it also fails when teacher samples come from a coarse solver. The two targets then differ in expectation, and the direction of the resulting difference between RSD and DAgger is an empirical question that the no-margin ablation addresses.

C.4 Decision Rules

The primary rule requires paired 95% item-bootstrap intervals above zero for RSD minus DAgger in pooled GR00T $\mathrm{KU}(S)$ and $\mathrm{CK}(S)$, with the lower endpoint for the $\mathrm{CL}(S)$ difference above -0.03. It compares complete recipes because DAgger uses sampled teacher actions and no alternative-name margin. Translation versus literal continuation is an operational reference because it compares S0 with a post-continuation checkpoint. HT, HO, and $\pi_{0.5}$ are KU-only replication checks; stratum and run summaries are descriptive. Secondary rules test online versus offline RSD in KU, full versus no-margin RSD in CL, gated injection in CK, positive decoding evidence, and directionally specific patching. The primary injection comparison conditions on one adapter run; Appendix I.5 reports three adapter seeds.

C.5 Paired Bootstrap

Algorithm 1 gives the interval computation for a method difference. The resampling unit is the item, every drawn index carries all instructions, layouts, repetitions, and seeds, and the same indices are applied to both methods before subtracting, since subtracting the endpoints of two separate intervals would not give the paired interval. Fixed resampling seeds make every reported interval recomputable when applied to the corresponding per-item selection records.

Algorithm 1 Paired item bootstrap for $\mathrm{KU}_A(T) - \mathrm{KU}_B(T)$.

Require: per-item contrasts $(\mathrm{CL}_i^A, \mathrm{CK}_i^A)$ and $(\mathrm{CL}_i^B, \mathrm{CK}_i^B)$ for $i \in T$, each averaged over 24 rollouts
1: **for** $j = 1, \ldots, 2000$ **do**
2: draw $|T|$ indices $i_1, \ldots, i_{|T|}$ from T uniformly with replacement
3: $\Delta_j \leftarrow \sum_m \mathrm{CK}_{i_m}^A / \sum_m \mathrm{CL}_{i_m}^A - \sum_m \mathrm{CK}_{i_m}^B / \sum_m \mathrm{CL}_{i_m}^B$
4: **return** point estimate $\mathrm{KU}_A(T) - \mathrm{KU}_B(T)$ and the 2.5th and 97.5th percentiles of $\{\Delta_j\}$

The pooled comparison resamples the union of the three splits and keeps their realized composition; it is not an unweighted mean of the split scores, because its effective weights depend on the selected-item counts and on the literal contrasts. Neither the pooled nor a within-split interval expresses uncertainty over unseen knowledge types, object categories, or simulator versions.

Each resampled item carries the eight rollouts from each of the three trained seeds, so the interval conditions on these runs and is not an interval over retraining. Treating the 24 rollouts per instruction as 24 independently trained policies would overstate replication. No significance test or multiple-comparison adjustment is applied to the secondary comparisons, which are interpreted descriptively.

C.6 Analytical Rollout-Noise Reference

Under the simplifying assumption that the 24 action-sampling outcomes for each item and instruction are conditionally independent given the fitted policies and fixed layouts, $\mathrm{Var}\,\hat{p}_i(x) \le 1/96$. If the two instructions of a contrast are also conditionally independent, $\mathrm{Var}\,\widehat{\mathrm{CK}}_i \le 1/48$ and $\mathrm{Var}\,\widehat{\mathrm{CL}}_i \le 1/48$. Under independence across items, their pooled means $\bar{K}$ and $\bar{C}$ each have a variance upper bound of $1/(48|S|)$. Ignoring covariance between numerator and denominator, the delta-method illustration is

$$\mathrm{Var}\,\widehat{\mathrm{KU}} \approx \frac{\mathrm{Var}\,\bar{K} + \mathrm{KU}^2\,\mathrm{Var}\,\bar{C}}{\bar{C}^2} \lesssim \frac{1 + \mathrm{KU}^2}{48\,|S|\,\bar{C}^2}. \tag{7}$$

If two methods also have independent action-sampling outcomes and similar $\bar{C}$ and KU, the corresponding approximate 95% half-width for their difference is

$$1.96\sqrt{\frac{2(1 + \mathrm{KU}^2)}{48\,|S|\,\bar{C}^2}}. \tag{8}$$

For $\bar{C} = 0.8$ and $\mathrm{KU} = 0.3$, the values are 0.052, 0.037, and 0.030 for $|S| = 100$, 200, and 300. The ± 0.04 band in Figure 3 is a rounded visual scale in this vicinity, *not* a resolution floor, confidence interval, or decision threshold. Shared environment initializations, correlated outcomes, unequal method-specific contrasts, covariance in the ratio, and between-item heterogeneity can change actual uncertainty. The primary interval uses the paired item bootstrap; the visual band does not replace its empirical uncertainty estimate.

D Training Specification

D.1 Initialization and Interfaces

The GR00T initialization is `nvidia/GR00T-N1.7-3B` (NVIDIA, 2026b) at public revision `2fc962b973bccdd5d8ce4f67cc63b264d6886495`, whose vision-language backbone is Cosmos-Reason2-2B (NVIDIA, 2026a) at revision `9ce19a195e423419c349abfc86fd07178b230561`. The public configuration selects backbone layer 16, uses a 40-action horizon, and integrates the flow with four steps; we set the selection index to 12 explicitly, and S0 trains the projection and action-side modules for this readout. The backbone is frozen explicitly instead of through default training flags, and normalization and fusion components outside it belong to the trained interface. Before its answers are treated as same-backbone knowledge, the language probe is compared tensor by tensor, including preprocessing, with the backbone loaded inside the policy.

The $\pi_{0.5}$ initialization is `lerobot/pi05_base` (Hugging Face LeRobot, 2026) at revision `b211f3d44c36b6acfcf7ae94a64e8e96f75a64ba`, with a 50-action chunk and ten integration steps, and its knowledge reference is `google/paligemma-3b-pt-224` at revision

35e4f46485b4d07967e7e9935bc3786aad50687c. Robot pretraining can change the prefix weights, so a shared tokenizer or architectural family does not make the source model a report of the policy's own knowledge.

For GR00T, evaluation uses deterministic center cropping consistent with the public image processing, with a 256-pixel target size and a 230-pixel crop; the $\pi_{0.5}$ interface uses 224×224 images. Each knowledge probe receives the processed view of the policy it accompanies, and random augmentation is restricted to training. The 14 valid action coordinates are embedded in 132 padded dimensions for GR00T and 32 for $\pi_{0.5}$, and all action losses mask the padded coordinates. Each checkpoint keeps its native horizon, and eight actions are executed before the next observation.

D.2 Run Matrix and Budget

The S0 dataset contains 10,240 literal demonstrations from 256 training layouts, each with all four objects as targets and ten demonstrations per target, so the observation alone never determines the requested object. Both architectures use the same layouts and embodiment. Each architecture has three S0 runs with seeds 0, 1, and 2, and every continuation run starts from the S0 run with the same seed. GR00T has seven continuation conditions and $\pi_{0.5}$ has three, which gives six S0 runs and $3(7 + 3) = 30$ continuation runs. Explicit translation reuses S0, and the injection experiment adds two shared adapters (Appendix E).

The core run matrix uses 364,000 updates: 120,000 for S0 ($6 \times 20{,}000$), 240,000 for continuation ($30 \times 8{,}000$), and 4,000 for the adapters. The 21 teacher-based continuation runs make 4,096 collection attempts each, 86,016 in total. Evaluation requires 46,080 selection episodes per model-method condition, so the 12 main conditions give 552,960 episodes, S0 screening adds 92,160, and the core injection analysis adds 138,240 plus 46,080 stored selections for the structural control. The extended E4-E9 controls are reported separately in Appendix I. Teacher sampling and translation differ across methods, so matching update and batch counts matches the optimization schedule rather than total compute.

Table 1: Training conditions. G and π mark conditions run for GR00T N1.7 and $\pi_{0.5}$. Every continuation condition starts from the S0 run with the same seed, uses 8,000 updates with batch size 32, and keeps 16 literal examples per batch whenever it uses a second data source.

Condition	Models	Supervision for indirect instructions	Margin	Comparison role
Literal continuation	G, π	none; literal demonstrations only	-	continuation control
Paraphrase BC	G	demo actions; 30% of instructions indirect	-	indirect exposure
Relabel-BC	G	success-filtered teacher rollouts	-	filtered teacher data
DAgger	G, π	teacher samples, student-visited states	-	sampled-action control
RSD	G, π	teacher velocity, student-visited states	✓	full method
Offline RSD	G	teacher velocity, fixed S0-visited states	✓	collection ablation
RSD w/o margin	G	teacher velocity, student-visited states	-	margin ablation
Explicit translation	G, π	test-time translator; no training	-	explicit reference

D.3 Training Settings

Table 2 lists the settings shared by all conditions. The batch of 32 contains 16 literal and 16 visited-state examples for methods with both sources, and 32 literal examples for literal continuation; paraphrase BC replaces 30% of instruction descriptions with descriptions from the indirect training pool, and Relabel-BC and DAgger keep the same literal anchor as RSD. Learning rates, loss weights, and gate thresholds are fixed before training, are not tuned on evaluation items, and are identical across the conditions that use them.

D.4 Collection and Teacher Supervision

The online buffer is refilled with 256 attempted episodes at update 1 and at every later 500-update boundary through update 7,501, which gives 16 refreshes and 4,096 attempts per run. A rejected translation consumes an attempt and contributes no states, each accepted episode records every replanning state until stable selection or timeout, and an empty buffer produces a literal-only update.

Table 2: Training settings shared by all conditions unless stated otherwise.

Setting	Value
S0 / continuation updates	20,000 / 8,000
Training seeds per condition	3 (0, 1, 2)
Batch size; literal / visited-state split	32; 16 / 16
GR00T S0 / continuation learning rate	10^{-4} / 4×10^{-5}
$\pi_{0.5}$ S0 / continuation learning rate	2.5×10^{-5} / 10^{-5}
Optimizer	AdamW, $\beta = (0.9, 0.95)$, weight decay 0.01
Gradient norm cap / warmup	1.0 / 500 updates
Continuation learning-rate decay	cosine to 0.1 of the initial rate
EMA decay	0.999
λ_d, λ_n, γ	1.0, 0.1, 0.5
Translation vote / margin gate	4 of 5 prompts / 0.15 nats per token
Endpoint dispersion scale	0.05 m
Buffer refresh / attempts per refresh	every 500 updates / 256
Teacher-control probability $\beta(n)$	$0.5 \max\{0, 1 - (n-1)/2400\}$
Paraphrase BC indirect fraction	0.30
GR00T / $\pi_{0.5}$ flow steps	4 / 10
GR00T / $\pi_{0.5}$ action horizon	40 / 50
Executed actions / control rate	8 / 20 Hz
GR00T readout layer	12 (public default 16)

At continuation update n, the teacher controls an episode with probability

$$\beta(n) = 0.5 \max\{0, 1 - (n-1)/2400\}. \tag{9}$$

The controller emits a full action chunk, of which eight actions are executed, and the observation and both instruction conditions are stored before the environment advances. During training the EMA teacher is queried at the stored state under the accepted literal name; after a collision or a large learner deviation it may be unreliable, and its predictions provide supervision without any guarantee of recovery.

Teacher confidence uses four action chunks sampled at the first observation of an episode: forward kinematics maps each chunk endpoint to an end-effector position, d is the root-mean-square distance of these positions from their mean, and the confidence is $\exp[-(d/0.05\,\mathrm{m})^2]$, whose product with the translation vote share defines $w(o, k)$. The weight is reused for every stored state of the episode and is a heuristic score, since low dispersion can also occur for a confidently wrong teacher.

Offline RSD collects its 4,096-attempt pool once, before continuation, by executing S0 on indirect instructions with the same gate, then applies the RSD velocity target and margin to these fixed states while querying the evolving teacher; this isolates adaptive collection at the level of the training recipe. Relabel-BC collects literal-teacher trajectories within the same attempt budget, keeps episodes ending in a stable lift of the translated object, and replaces their instructions with the paired indirect descriptions, a filter that uses training-episode outcomes a filter-free learner does not receive. DAgger uses the gate, schedule, weight, and budget of RSD, fits the teacher's sampled actions with flow-matching behavior cloning, and has no margin.

D.5 Flow Adapters and Losses

For the π_0 family, native interpolation is $x_t = t\epsilon + (1-t)a$ with native target $\epsilon - a$, so canonical time is $s = 1 - t$ and the canonical velocity is the negative of the native one; GR00T already uses the noise-to-action direction with its own discrete time encoding. Each adapter calls the native training-time sampler and converts its output, because a Beta parameter in a configuration file cannot be interpreted without the subsequent time transformation. For an action tensor with horizon H, padded dimension D_{pad}, and mask $M_d \in \{0, 1\}$, the masked distance is

$$D_M(u, v)^2 = \frac{\sum_{h=1}^{H} \sum_{d=1}^{D_{\mathrm{pad}}} M_d (u_{hd} - v_{hd})^2}{H \sum_{d=1}^{D_{\mathrm{pad}}} M_d}. \tag{10}$$

Padding contributes to neither velocity matching nor the margin. The teacher target, the alternative velocity, and the margin scale are detached from differentiation. EMA updates follow each student optimizer step and apply to trained parameters only.

Three checks preceded training: the adapter's action sample matched the native sampler for identical noise, model state, and integration schedule; converted native time draws reproduced the native time distribution; and perturbing only padded coordinates left every loss unchanged. Every trained condition passed these checks.

E Fictional-Fact Injection

E.1 Adapters and Conditions

The injection experiment assigns fictional manufacturers to 24 object identities and writes five references per identity, which gives 120 evaluation items. Each fictional manufacturer identifies exactly one object within an item, and none of the facts can have appeared in backbone pretraining. The adapter training data contain 30 rendered views per object and 20 question templates per view, for 14,400 VQA examples. The adapter objective contains no actions or demonstrations. Evaluation views and question templates are disjoint from the training examples.

Both adapters are trained on the same frozen GR00T backbone, the lower one on language layers 1-12, at or below the readout, and the upper one on layers 13-28. Each uses LoRA (Hu et al., 2022) with rank 16, scale 32, and dropout 0.05 on the query and value projections only, trained for 2,000 updates at batch size 32 with AdamW at learning rate 2×10^{-5} and seed 0. Both adapters are shared across the literal-continuation, DAgger, and RSD policies, and the explicit route loads the matching adapter when it translates. A third condition supplies the fact in the instruction without any adapter: "The [object name] is made by [fictional manufacturer]. Pick up the object made by [fictional manufacturer] and place it in the tray." This context condition supplies the answer name along with the fact, so it tests access to a fact presented in context rather than one stored in weights.

E.2 Gate and Reporting

The injection counts as successful only if the explicit-name probe and the location probe both reach 0.90 accuracy on the 120 items, using the templates of Appendix B with the fictional reference in place of `{k}`. If either probe misses the gate, the condition stays in the analysis and counts as incomplete injection or incomplete composition of the fact with the scene; a high explicit-name probe alone does not qualify it as successful knowledge acquisition. For every method and condition we estimate CK with its 95% item-bootstrap interval, CL, and the change in CK relative to the unadapted policy on all 120 items; Figure 6 displays the knowledge contrasts. KU is secondary because the injected items are not screened by S0.

E.3 Structural Interpretation

When only layers above the readout are modified, the direct action computation receives no changed hidden state, provided that no output of those layers feeds back into its inputs, caches, normalization, or preprocessing; with identical noise and evaluation state its output must then equal the unadapted output, which we verify episode by episode over all 11,520 selections of each direct method. The check concerns the effect of the adapter and does not require the original CK to be zero, since finite samples and instruction preferences unrelated to the fact can already produce a nonzero contrast. The explicit route can change, because its language head reads the adapted layers and can pass a different literal name to the policy.

The lower adapter changes representations used for both literal and indirect action, so its effect combines access to the fact with changes in the representation, which is why CL is reported with CK. If it fails the dual-probe gate, a weak knowledge contrast cannot be attributed to the action head and the supported conclusion is limited transfer under incomplete injection; if it passes, a knowledge contrast for RSD above that of DAgger indicates that the distilled mapping extends to facts introduced after continuation training.

F Representation Analyses

F.1 Readout Probe

E1 measures the linear decodability of the referenced object's slot from the representation read by the action head. For GR00T the features are the layer-12 hidden states at the instruction-token positions; for $\pi_{0.5}$ they are the final-layer prefix states, which the action expert reads through its attention to the prefix. Image-token states are excluded, so the probe sees only what the instruction tokens have gathered from the image. Two feature vectors are formed per episode, the mean over the reference span and the mean over all instruction tokens; the span features are primary.

The label is the slot, one of four, that holds the referenced object in the current layout; the cyclic layouts place each object in each slot once, so the label cannot be predicted from object identity or instruction text alone. The probe is a multinomial logistic regression with an L_2 penalty chosen by five-fold cross-validation with folds grouped by object tuple, trained on the 360 indirect training items in all four layouts under both instruction types and evaluated on every evaluation item. The backbone is frozen, so one probe per architecture serves all training conditions.

Two controls bound what the probe can read. In the control task (Hewitt & Liang, 2019), each object identity receives a random but fixed slot label and the probe is retrained on these labels with the same features and regularization; a probe that exceeds the control accuracy decodes structure beyond memorized identities, and selectivity is the indirect-instruction accuracy minus the control accuracy. In the shuffled-image control, the instruction states are computed with the image of a different layout while the labels keep the original one, so accuracy near 0.25 shows that the probe reads scene-grounded information rather than a textual cue.

For each item, the indirect-instruction accuracy is its share of the four layouts in which the probe names the correct slot, and we report mean accuracy and selectivity on S, on each split within S, and on all 480 items with 95% item-bootstrap intervals. We also report the item-level correlation between indirect-instruction accuracy and CK_i for literal continuation, DAgger, and RSD. This observational association may reflect item difficulty and does not show that the decoded feature drives the action.

F.2 Activation Patching

E3 measures the selection change produced by replacing the reference span at the action-head readout. It runs on GR00T, whose action head reads a single backbone layer, for the literal-continuation and RSD policies of all three seeds. Patching modifies the layer-12 states the action head receives, leaves earlier layers and image-token states unchanged, and is applied at every replanning call. Layouts, repetitions, seeds, and noise follow the main evaluation, so each patched episode has an unpatched counterpart. Only the two indirect instructions are patched, so every condition keeps the literal contrast of its unpatched policy.

Four conditions are evaluated. P0 replaces the states of the whole indirect instruction by those of the literal instruction on the same observation. Because the image-token states are left unchanged and the sequence length may differ, this is an approximate literal-route pipeline check rather than guaranteed identity with an unpatched literal run. A departure from literal selection may reflect token alignment or instruction-dependent image states as well as an implementation error; comparisons of readout tensors on the same observation distinguish these possibilities. P1 replaces only the reference span by the span of the object name from the literal instruction, leaving the surrounding tokens and changing the sequence length with the span length. P2 adds an item-agnostic steering vector to every token of the reference span, the mean difference between name-span and reference-span states over the indirect training items, rescaled to the average norm of that difference. P3 applies the P1 replacement with the alternative name.

KU is computed on S for each condition, and the share of the gap recovered is its KU minus the unpatched KU divided by the P0 KU minus the unpatched KU, estimated with a paired item bootstrap that resamples the same items for all three terms. For P3 we report $-\mathrm{CK}/\mathrm{CL}$, the degree to which selection follows the patched name. A P3 effect opposite in sign to P1 supports content sensitivity, but span replacement also changes sequence length and downstream token positions. Thus P1 exceeding P2 is compatible with object-specific information; it does not isolate that information from length and alignment effects without a length-matched patching control (Appendix I.6).

F.3 Analysis Scope

A positive probe result shows that the referent can be decoded linearly from the readout states, not that the action head uses the same features or that a nonlinear reader could not decode more. The patching conditions test sufficiency of the reference-span states for restoring selection, not that the unpatched policy fails for the same reason on every item, and they test no other position or layer. Both analyses run on the frozen backbones of the main protocol.

G Additional Results

This appendix reports the injection and representation analyses that complement Section 6. The injection comparisons use 120 items, and the representation analyses use the conditioning sets defined in Section 5.

G.1 Injection and Representation Outcomes

The context and lower-adapter conditions pass both probe thresholds: the lower adapter reaches 0.925 name accuracy and 0.908 location accuracy. The upper adapter remains below the gate and leaves direct-route selections unchanged. Under the successful lower injection, RSD reaches $\mathrm{CK} = 0.185$ $[0.148, 0.222]$, compared with 0.122 $[0.086, 0.158]$ for DAgger and 0.041 $[0.013, 0.070]$ for literal continuation.

Probe selectivity on S is 0.250 $[0.197, 0.304]$ for GR00T N1.7 and 0.171 $[0.103, 0.236]$ for $\pi_{0.5}$. For RSD, item-level decoding-contrast correlations are 0.33 $[0.18, 0.46]$ and 0.30 $[0.13, 0.45]$, relating decodability to behavior without treating it as a complete explanation.

Replacing the reference span recovers 0.64 of the KU gap for literal continuation $[0.60, 0.68]$ and 0.66 for RSD $[0.61, 0.71]$. A generic steering vector recovers only 0.08 and 0.10, while an alternative name drives KU to -0.65 and -0.59, demonstrating a larger, directionally specific effect.

H Item-Level Distributional Analysis

This appendix reports item-level summaries behind the pooled GR00T RSD-DAgger comparison. The paired analysis uses the fixed 157-item conditioning set, the four matched instructions, and 24 rollouts per instruction. Table 3 gives pooled contrasts, Appendix H.5 gives effect-sign counts and outcome summaries, and Appendix H.6 states the scope of the item-bootstrap interval. Figure 7 is a separate mathematical example about non-identifiability from aggregate means.

H.1 Pooled Contrast Components

Table 3 collects the pooled GR00T contrasts on S. Values are rounded to three decimals; dividing the displayed CK by the displayed CL can differ slightly from KU calculated before rounding. The table separates a change in knowledge contrast from a ratio change caused by weaker literal control.

Table 3: Pooled contrasts on the fixed GR00T set S ($|S| = 157$). A positive literal contrast means the target is selected more often when named than when the alternative is named.

Condition	$\overline{\mathrm{CL}}$	$\overline{\mathrm{CK}}$	$\mathrm{KU}(S)$
Literal continuation	0.751	0.075	0.100
Paraphrase BC	0.710	0.125	0.176
Relabel-BC	0.732	0.139	0.190
DAgger	0.729	0.153	0.210
RSD without margin	0.736	0.198	0.269
Offline RSD	0.736	0.197	0.268
RSD	0.740	0.230	0.311
Explicit translation	0.736	0.633	0.860

H.2 Episode Records and Validation

The selection outcome is recorded per rollout with model, method, training seed, item, split, knowledge type, layout, repetition, instruction key, target, alternative, selected object, termination reason, and stable-lift tick. The instruction keys l, l', k, k' identify the target and alternative literal and indirect references. A selected identity may be the target, alternative, either distractor, or null for no lift or ambiguous simultaneous lift. Item metadata record the four objects, their category and visual attributes, both references, predicate source, and ambiguity checks. Probe records contain the six prompt-level name and location votes and margins, the five translation votes, and S0 screening contrasts. Checkpoint metadata specify the model revision, readout layer, optimizer settings, evaluation code revision, and artifact hashes.

Each model-method condition contains $480 \times 4 \times 4 \times 2 \times 3 = 46{,}080$ selection episodes, one for each item, instruction, layout, repetition, and training seed. Each $\hat{p}_i(x)$ averages 24 episodes. Target and alternative instructions share layout and repetition identifiers, and S is held fixed across methods within a model. No-lift and ambiguous-lift episodes remain in the denominators. The primary selection score is the first stable lift; approach, contact, grasp, transport, and placement annotations support the error-stage analysis in Appendix I.3.

H.3 Distributional Estimation

We compare item-level CL_i, CK_i, and the paired effect $\Delta\mathrm{CK}_i = \mathrm{CK}_i^{\mathrm{RSD}} - \mathrm{CK}_i^{\mathrm{DAgger}}$ on the same set. Each estimate has $1/24$ resolution, so effect-sign counts and summary statistics retain information lost in a pooled mean. The pooled ratio is $\mathrm{KU}(S) = \sum_i \mathrm{CK}_i / \sum_i \mathrm{CL}_i$; per-item ratios $\mathrm{CK}_i/\mathrm{CL}_i$ cannot replace this estimator, particularly when CL_i is near zero. A KU score over all 480 items measures a different population from the screened comparison; TE, HT, and HO retain their selected-item denominators rather than being equally weighted.

H.4 Aggregate Non-Identifiability

The measured RSD means on S are approximately $\overline{\mathrm{CL}} = 0.740$ and $\overline{\mathrm{CK}} = 0.230$, giving $\mathrm{KU} = 0.311$. To show why these means do not determine the distribution, consider two *mathematical examples*, each with 157 items and $\mathrm{CL}_i = 0.740$ for every item. In example A, every item has $\mathrm{CK}_i = 0.230$. In example B, 40 items have $\mathrm{CK}_i = 0.700$ and 117 have $\mathrm{CK}_i = (157 \cdot 0.230 - 40 \cdot 0.700)/117 \approx 0.069$. Both reproduce the same means and KU within rounding, although one is diffuse and the other concentrated. These contrast assignments are a mathematical counterexample, separate from the observed item summaries below.

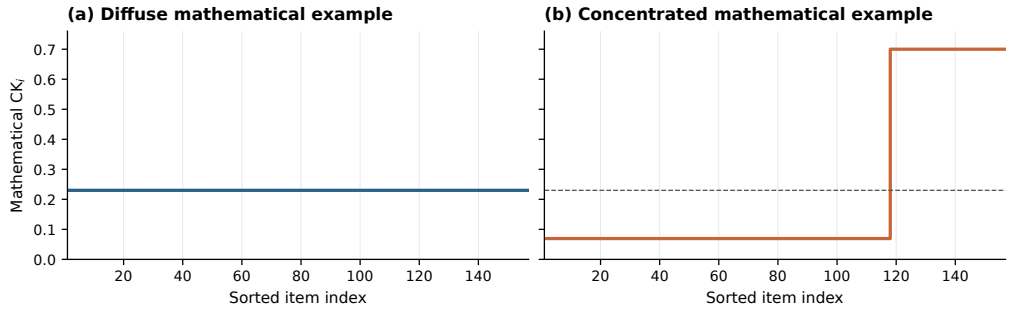


Figure 7: Mathematical illustration of aggregate non-identifiability. Two distinct per-item CK distributions yield $\overline{\mathrm{CK}} = 0.230$ across 157 items. With $\mathrm{CL}_i = 0.740$ throughout, both give $\mathrm{KU} \approx 0.311$. Neither curve depicts observed item outcomes.

H.5 Item-Level RSD-DAgger Effects

The GR00T comparison covers 71 TE, 46 HT, and 40 HO items, with 24 outcomes under each of the four instructions for each method. DAgger has $(\overline{\mathrm{CL}}, \overline{\mathrm{CK}}, \mathrm{KU}) = (0.729, 0.153, 0.210)$; RSD has $(0.740, 0.230, 0.311)$. Split KUs appear in Figure 3. The S0 screen retains 157 of the 480 items; 161

fail the literal-control threshold, 81 have unknown explicit answers, and 81 have confidently wrong answers.

Paired $\Delta\mathrm{CK}_i$ is positive on 110 items, negative on 35, and zero on 12. The split sign counts (positive/negative/zero) are 52/14/5 for TE, 32/10/4 for HT, and 26/11/3 for HO. There are 12 RSD items and 29 DAgger items with $\mathrm{CK}_i \leq 0$. Median $\Delta\mathrm{CK}_i$ is 0.0833 and mean $\Delta\mathrm{CK}_i$ is 0.0772; the largest 10% of positive-gain items contribute 24% of positive gains. The paired item-bootstrap interval for $\Delta\mathrm{KU}$ is $[0.067, 0.135]$. These numbers show a positive pooled effect alongside item-level heterogeneity.

Table 4: Outcome distribution on the 157 GR00T items (percent of 3,768 rollouts per method-instruction row). Distractor combines the two distractor identities. Rounding can make a row sum to 99.9 or 100.1; no ambiguous-lift cases occurred.

Method	Instruction	Target	Alternative	Distractor	No lift
DAgger	k	30.2	41.9	14.5	13.4
DAgger	k'	14.9	49.8	19.0	16.3
RSD	k	38.0	34.5	14.5	13.0
RSD	k'	15.0	50.5	17.5	17.0

The pooled CK increase chiefly reflects the higher target rate under k (30.2% to 38.0%); the target rate under k' is nearly unchanged. No-lift under k changes by less than one percentage point, so a large reduction in timeouts does not explain the pooled contrast gain. Stable-lift selection still combines referent choice and grasp execution; the error-stage audit in Appendix I.3 addresses these stages separately.

H.6 Uncertainty across Items and Runs

The primary paired bootstrap resamples items while retaining the four instructions and all rollouts from the three trained runs for each selected item. It estimates sensitivity to the composition of the fixed screened set conditional on those runs. It does not quantify the variation that would arise from retraining the models. Individual seed effects in Figure 5 show run variation descriptively; with only three training seeds, a retraining interval would be unstable. The 24 rollouts per item reflect repeated evaluations of fitted policies, not independent policy replicas.

I Extended Experiments

We report same-checkpoint routing, factorial training controls, item-level and error-stage analysis, task shift, adapter replication, and length-matched patching. These experiments refine the interpretation of the pooled RSD-DAgger comparison; they do not turn a difference between complete training recipes into an isolated effect of one loss term.

I.1 E4: Same-Checkpoint Explicit-Route Gap

We evaluated each frozen checkpoint twice on matched items, layouts, repetitions, and action-noise seeds: once with the indirect instruction and once with its gated translation followed by the same checkpoint under the literal instruction. The comparison covers S0, literal continuation, DAgger, RSD without margin, and RSD. The translator, gate, candidate names, and fallback rule were fixed across checkpoints, without test-time access to the correct target. We recorded correct acceptance, wrong acceptance, and rejection separately. Both routes share their literal-instruction outcomes, so their CL denominators are identical.

For checkpoint m, define $\Delta^m_{\mathrm{route}} = \mathrm{KU}_{\mathrm{translate}(m)}(S) - \mathrm{KU}_{\mathrm{direct}(m)}(S)$ and $\Delta\mathrm{CK}^m_{\mathrm{route}} = \mathrm{CK}_{\mathrm{translate}(m)}(S) - \mathrm{CK}_{\mathrm{direct}(m)}(S)$. Paired item resampling retains matched environment initializations, and the primary E4 estimand is the S0 CK difference on the original screen. At S0, direct CK is 0.081 and translated CK is 0.633, a difference of 0.552 with common CL 0.736. Their KU values are 0.110 and 0.860, a route gap of 0.750. Translation accepts a correct name in 86% of episodes, a wrong name in 4%, and rejects 10%. This same-checkpoint gap removes the training mismatch in

the headline operational reference, but translation invokes a different language computation. It does not by itself show that the direct action readout could use the fact.

I.2 E5: Factorial Loss Ablation

We crossed action-sample versus teacher-velocity targets with absence versus presence of the alternative-name margin. All four arms started from seed-matched S0 checkpoints and shared 8,000 updates, the literal anchor, translation gate, EMA rule, collection-attempt schedule, teacher-control schedule, confidence weight, and batch composition. The sampled-target-plus-margin arm computed its margin from its own EMA teacher, as RSD did. The online arms can still visit different states and learn different teacher parameters, so their contrasts describe the trained recipes.

Let $K_{v,1}$, $K_{v,0}$, $K_{a,1}$, and $K_{a,0}$ denote pooled KU for velocity and action-sample targets with and without margin. The four observed $(\overline{\mathrm{CL}}, \overline{\mathrm{CK}}, \mathrm{KU})$ cells are $(0.729, 0.153, 0.210)$ for sample/no margin, $(0.733, 0.171, 0.233)$ for sample/with margin, $(0.736, 0.198, 0.269)$ for velocity/no margin, and $(0.740, 0.230, 0.311)$ for velocity/with margin. The no-margin velocity-sample contrast is $\Delta\mathrm{KU} = 0.059$, with $\Delta\mathrm{CK} = 0.045$ and $\Delta\mathrm{CL} = 0.007$. Using rounded cells, the target main effect is 0.069, margin main effect 0.033, and interaction 0.019 in KU. These are contrasts of pooled ratios, not averages of per-item ratios.

We also trained a fixed-buffer comparison from each S0 seed. A predeclared S0 indirect controller mixed with its literal teacher supplied stored states independently of the four factorial arms. States, accepted translations, confidence weights, a common teacher snapshot, noise draws, and flow-time draws were fixed across students; the four students trained without EMA teacher updates. Relative to sampled-action cloning, the velocity target raised CK by 0.040 and CL by 0.003. At updates 0, 2,000, 4,000, and 8,000, we measured 32 minibatch gradients per target on fixed stored states with common indices, including covariance trace, gradient norm, cosine similarity to the minibatch mean, and teacher-sample residual. The sample-gradient covariance trace was 1.60 times the velocity-gradient trace. The diagnostic is consistent with lower gradient variance under the velocity target; the online KU difference can additionally reflect state visitation and teacher evolution.

I.3 E6: Distribution and Error-Stage Analysis

We computed paired per-item CK effects on the fixed GR00T set as described in Appendix H.5. Median $\Delta\mathrm{CK}_i$ is 0.0833, 110 of 157 items improve, and the largest 10% of positive-gain items account for 24% of positive gains. The denominator for that share is $\sum_i \max(0, \Delta\mathrm{CK}_i)$; negative effects are retained separately in the net effect. Table 3 reports the pooled CK and CL components.

The error-stage analysis distinguishes reference choice, approach, contact, lift, and placement using a deterministic state-machine scorer with thresholds fixed before method comparisons. A stratified, method-blinded video audit covers at least 50 episodes per method, split, and instruction type. Difficulty strata use pre-treatment S0 literal contrast, probe margin, location-probe correctness, predicate type, and object frequency in S0 demonstrations. On HO, object frequency in training is zero, so visual and language complexity define the relevant strata. Stage and difficulty analyses are descriptive where paired component intervals are unavailable; the pooled KU result should not be read as a pure referent-choice measure because stable lift also requires grasp execution.

I.4 E7: Six-Object and Two-Step Shift

We evaluated the original three trained checkpoints without adapting them to 120 new six-object scenes. Each scene has unique target and alternative references, six cyclic layouts that place every object in each slot once, two additional distractors, four instructions, and two repetitions. Object tuples, templates, and scene seeds are disjoint from the four-object benchmark. This gives $120 \times 4 \times 6 \times 2 \times 3 = 17{,}280$ selection episodes per method. A two-step task additionally requires placing the referenced object in one of two named receptacles; first stable lift and final placement are scored separately.

The six-object S0 screen uses a six-candidate probe and the original literal-contrast threshold of 0.6, and retains 58 of 120 items. On this set, DAgger has $(\overline{\mathrm{CL}}, \overline{\mathrm{CK}}, \mathrm{KU}) = (0.680, 0.116, 0.171)$ and RSD has $(0.679, 0.170, 0.250)$, giving $\Delta\mathrm{CK} = 0.054$. The CK comparison uses a CL guardrail

whose paired lower endpoint must exceed -0.03. Placement is 0.24 for DAgger and 0.32 for RSD. These point estimates extend the selection pattern to a six-object evaluation shift; the placement comparison is descriptive because a separate interval is not reported. Evaluating the same checkpoints under shift does not add retraining replication.

I.5 E8: Injection and Readout Replication

We trained three independent lower-adapter seeds using independently shuffled VQA examples for the same 24 fictional mappings. Each seed was evaluated against the 0.90 dual name-and-location probe gate on all 120 injected items, without screening by action outcome. All three passed. The RSD-DAgger CK differences were 0.060, 0.058, and 0.061. The main injection analysis in Section 6 uses one adapter seed; this replication shows the direction across three adapter trainings for the fixed mapping set.

We also repeated the GR00T analysis with its public layer-16 readout, its own S0 adaptation, and matched continuation conditions. The public-readout screen contains 142 items. DAgger KU is 0.197 and RSD KU is 0.286, a gap of 0.089. The layer-12 and layer-16 screens can contain different items, so the within-readout contrasts are the primary comparisons; a common-screen intersection is a sensitivity comparison. Differences across readouts can reflect both representation access and retrained action interfaces.

I.6 E9: Length-Matched Patching

P1 and P3 replace reference spans with object-name spans, which can have different token counts, whereas P2 preserves the original span length. We repeated P1 and P3 on the 73-item subset with identical source and replacement token counts, preventing a downstream position shift; P0 remains the whole-instruction pipeline control. On this subset, target-name patch recovery is 0.60, alternative-name following is 0.55, and a length-matched unrelated-name patch recovers 0.12. The unrelated name was chosen independently of action outcomes and balanced over object frequency. The difference between target-name and unrelated-name patching supports content specificity on this subset, although patching still intervenes on representations rather than isolating every internal causal step.

I.7 Summary of Extended Experiments

Table 5 collects the observed results and the interpretation supported by each comparison. The same-checkpoint route comparison is diagnostic, and the online factorial remains coupled to teacher and state changes. Item-level, shift, adapter, readout, and patching results specify where the pooled RSD result does and does not extend.

Table 5: Extended experimental results. Values are measured on the indicated checkpoints and evaluation sets; interval and replication limits are stated in the corresponding sections.

Experiment and Estimand	Observed Result	Interpretation
GR00T RSD-DAgger ΔKU	0.101	Paired item interval $[0.067, 0.135]$
E4 same-checkpoint S0 route ΔCK	0.552	Direct CK 0.081, translated CK 0.633; common CL 0.736
E5 no-margin velocity-sample ΔKU	0.059	CK $+0.045$, CL $+0.007$; online factors remain coupled
E6 per-item ΔCK distribution	$110/35/12$	Positive/negative/zero items among 157
E7 six-object RSD-DAgger ΔCK	0.054	Screen retains 58 of 120; placement 0.32 vs 0.24
E8 lower-adapter CK gaps, three seeds	$0.060, 0.058, 0.061$	All three dual probes meet the 0.90 gate
E8 public layer-16 ΔKU	0.089	142 screened items; the within-readout gap is positive
E9 matched-length P1 vs unrelated recovery	0.60 vs 0.12	73 items; P3 alternative following 0.55