RoboSemanticBench: Diagnosing Semantic Grounding in Action Prediction for VLA Models

Anonymous ACL submission

Abstract

Vision-language-action (VLA) models are built on the premise that semantic understanding from pretrained language or vision-language backbones should guide robot action prediction. Yet robot fine-tuning is optimized as imitation over task-specific action distributions, and many evaluations can be solved through visual or instruction-action shortcuts. We introduce **RoboSemanticBench (RSB)**, an embodied benchmark for diagnosing *semantic grounding in action prediction*: whether post-trained VLA models can use complex instruction semantics to select and manipulate the correct physical target. In each episode, a robot receives a multiple-choice math or general-knowledge question, observes candidate answer blocks, and must grasp the block corresponding to the correct answer. RSB covers controlled arithmetic, grade-school mathematical understanding, and commonsense or factual understanding under four-choice and ten-choice suites. Across representative VLA models, we find that many policies learn to grasp candidate blocks but select the semantically correct block at near-random or below-random rates after controlling for grasp success, revealing a persistent gap between backbone-level semantic competence and action prediction. The benchmark code, generated data, and evaluation scripts are released at https://anonymous.4open.science/r/RoboSemanticBench-F4BD.

1 Introduction

Vision-language-action (VLA) models are motivated by a compelling promise: the semantic competence of pretrained language or vision-language models should become part of robot action prediction. Representative systems, including π_0, are often described as dual-system architectures with a low-frequency System-2 *Semantic Expert* and a high-frequency System-1 *Action Expert* (Kim et al., 2024; Black et al., 2024; Zhou et al., 2025). As

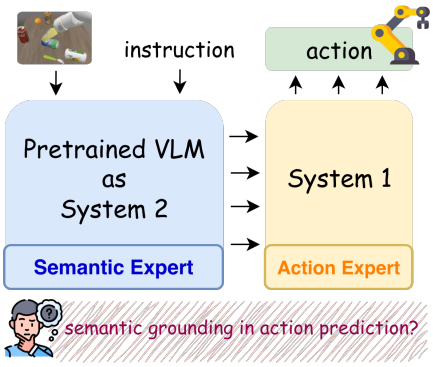

Figure 1: Dual-system VLA architecture and the semantically grounded action prediction question. RoboSemanticBench tests whether the System-2 Semantic Expert's competence is preserved and used by the System-1 Action Expert after VLA post-training.

illustrated in Figure 1, the Semantic Expert processes observations and instructions, while the Action Expert turns its outputs and proprioception into continuous actions. Under this view, a VLM backbone should not merely condition a controller on text; its general knowledge and semantic understanding should participate in the action-generation pathway.

The central concern is that this promise may not survive current VLA post-training pipelines. Robot demonstrations are much smaller and more task-specific than language pretraining corpora, and imitation losses can reward fitting conditional action distributions even when the Semantic Expert is weakened or decoupled from the Action Expert. In a typical demonstration dataset, an instruction is paired with a successful trajectory, but the loss rarely forces the model to expose the semantic decision that made the trajectory correct. The policy may therefore learn instruction-action or visual shortcuts rather than use language semantics to determine which action is correct, creating a direct tension with the motivation for VLA models (Han-

cock et al., 2025; Zhang et al., 2026a, 2025b).

This tension matters because realistic robot instructions are often complex, underspecified, or require commonsense interpretation, whereas many simulation benchmarks use short explicit commands such as picking a named object or moving to a visible location. In such settings, VLA policies can ignore language and still score well by exploiting visual shortcuts or dataset regularities (Lian et al., 2026; Fang et al., 2026; Guo et al., 2026). A high task success rate can therefore be ambiguous: it may indicate genuine instruction understanding, or merely that the benchmark admits non-semantic shortcuts. Recent studies further suggest that preserving the general semantic competence of pretrained VLM backbones is important for action-generation generalization (Yu et al., 2026; Zhang et al., 2026b; Guo et al., 2026). Yet existing benchmarks rarely measure whether such competence is actually grounded in action prediction. We call the missing capability ***semantic grounding in action prediction***: when controlling a robot, a VLA model should understand the semantics of a human instruction, ground them in the current observation, and act according to the user's intent.

We propose **RoboSemanticBench** (**RSB**), a controlled diagnostic for semantic grounding in action prediction. As shown in Figure 2, each episode asks a VLA to understand a multiple-choice math or general-knowledge instruction, bind the correct answer to a visible target, and execute the corresponding grasp. Because the manipulation primitive is fixed while the semantic content and option set vary, RSB tests whether instruction semantics guide target selection rather than merely fitting instruction-action correlations.

This design is deliberately simple in the motor domain but demanding in the semantic domain. It exposes two questions: whether the Semantic Expert can still solve the instruction-level problem, and whether the Action Expert can follow the resulting implicit target during action prediction. A model that can reliably grasp candidate targets should still fail if it cannot ground the instruction's semantic answer into the target-selection action. We therefore compare grasping any candidate block with grasping the correct one: a large GSR–TSR gap reveals successful manipulation but failed semantic grounding. This makes RSB a diagnostic of the VLA interface between semantic understanding and action prediction, rather than a generic test of grasping skill.

Our contributions are:

- We introduce **RoboSemanticBench**, a benchmark that turns math, hard-math, and general-semantic understanding into embodied answer selection across six evaluation suites.
- We define GSR, TSR, and nSG to separate low-level grasping from semantic target selection and diagnose whether instruction semantics participate in action prediction.
- We evaluate representative VLA models and show that many perform near or below random semantic target selection after controlling for grasp success.
- We report negative interventions and error analyses, showing that CoT-style ReasoningVLA and VLA cotraining do not reliably close the semantic grounding gap.

2 Related Work

2.1 Vision-Language-Action Models and Benchmarks

Recent VLA models connect pretrained vision-language representations with robot control. Early embodied foundation models show that web-scale visual-language pretraining can provide reusable semantic knowledge for robot tasks (Driess et al., 2023; Brohan et al., 2023); later systems scale this idea through large robot datasets and open generalist policies, including Open X-Embodiment/RT-X (O'Neill et al., 2024), Octo (Team et al., 2024), OpenVLA (Kim et al., 2024), π_0 (Black et al., 2024), $\pi_{0.5}$ (Intelligence et al., 2025), and GR00T N1 (NVIDIA et al., 2025). Recent work further studies how to adapt VLM backbones into VLA policies without losing pretrained capability (Hancock et al., 2025; Yu et al., 2026; Zhang et al., 2026b; Guo et al., 2026). Yet standard task success does not reveal whether semantic competence is actually preserved and used for action prediction.

Benchmarks for language-conditioned manipulation measure complementary abilities. CALVIN and LIBERO study multi-task and lifelong manipulation (Mees et al., 2022; Liu et al., 2023), SimplerEnv evaluates policies through simulation-based reproduction (Li et al., 2024), and other suites scale household tasks, robot data generation, world-knowledge manipulation, or VLA comparison (Nasiriany et al., 2024; Zhang et al., 2024;

Chen et al., 2025; Zhang et al., 2025a; Choi et al., 2026). These benchmarks are valuable for robustness and generalization, but their task success metrics often entangle motor execution, object recognition, and language grounding. RoboSemanticBench instead fixes the manipulation primitive and varies the instruction's semantic content, making it a targeted diagnostic for whether semantic understanding is grounded in robot action prediction.

2.2 Diagnosing Language Grounding and Shortcut Behavior in VLAs

Several studies question whether VLA policies use language as intended. Policies may exploit visual regularities or action priors, ignore instructions under conflicting visual evidence, or suffer degraded VLM representations after robot fine-tuning (Yu et al., 2026; Fang et al., 2026; Guo et al., 2026; Hancock et al., 2025; Zhang et al., 2026a, 2025b). Related diagnostics probe instruction perturbations, counterfactual commands, linguistic diversity, and distribution shift (Hou and Zhao, 2026; Fang et al., 2025; Zhang et al., 2026c; Wanna et al., 2026). RoboSemanticBench complements them by making the instruction itself a semantic problem: the policy must answer a math or general-knowledge question, bind the answer to an observed target, and execute the grasp. This evaluates *semantic grounding in action prediction* rather than language sensitivity alone.

3 RoboSemanticBench

Motivation. Most embodied benchmarks are designed to evaluate manipulation robustness, task generalization, or policy transfer, but their language instructions are often semantically simple. For example, LIBERO, SimplerEnv, and RoboTwin benchmarks commonly construct evaluation instructions from predefined language templates, such as short object-selection or placement commands (Liu et al., 2023; Li et al., 2024; Chen et al., 2025). These instructions are useful for controlled manipulation evaluation, but they cover a narrow semantic range and rarely require deeper commonsense, arithmetic, or problem-solving semantics. As a result, they do not fully reflect the architectural promise of the VLA paradigm, where a pretrained Semantic Expert is expected to contribute rich language understanding to action generation, and they differ substantially from realistic human instructions that are often diverse, underspecified, and semantically loaded. RoboSemanticBench is designed to fill this diagnostic gap by making semantic understanding a necessary condition for selecting the correct physical action.

3.1 Benchmark Overview

RoboSemanticBench evaluates whether semantic decisions made from language are used to generate robot actions. Each episode contains a question q, candidate options $\mathcal{O} = \{o_1, \ldots, o_N\}$, visible answer blocks $\mathcal{B} = \{b_1, \ldots, b_N\}$, and an option-to-block mapping $m : \mathcal{O} \rightarrow \mathcal{B}$. The policy receives an instruction containing q and the mapping, observes the scene, and succeeds only if it moves the block associated with the correct answer into the answer zone.

The physical action is always a single answer-selection primitive: pick the selected candidate block and place it in the gray answer zone. The semantic content changes across episodes, requiring the policy to solve the question, compare options, ground the chosen option to the visible block, and execute the pick-and-place action. Importantly, the correct target is not tied to a fixed color, letter, position, or trajectory; it is determined by the question and the episode-specific option mapping. This isolates whether instruction semantics participate in action prediction.

3.2 Semantic Task Construction

RSB contains three semantic subsets with different semantic demands.

(i) RSB-Math uses procedurally generated arithmetic questions. Each question samples one of three controlled forms: two-digit addition, two-digit subtraction, or one-digit by two-digit multiplication. The correct numerical answer is mixed with nearby distractors, so the policy must compute the result rather than rely on superficial option patterns.

(ii) RSB-HardMath uses grade-school word problems derived from GSM8K (Cobbe et al., 2021). The problem text is used as q, the final answer is the correct option, and distractors are taken from prepared option fields or generated around the answer when needed. Unlike RSB-Math, these questions require extracting quantities from natural-language context and composing multiple relations before choosing an answer. This subset therefore tests whether a VLA policy can

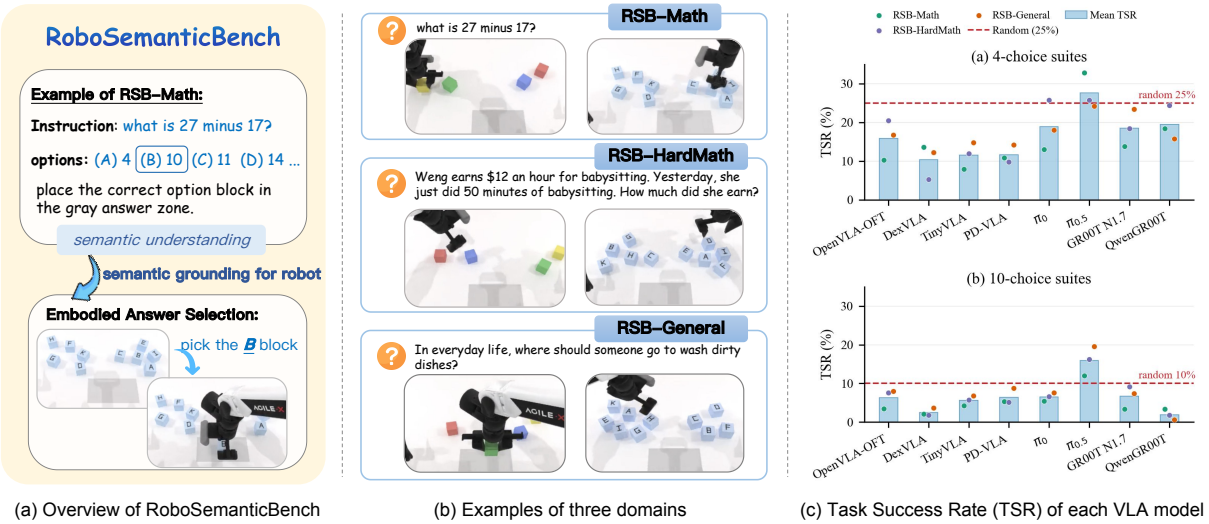

Figure 2: Overview of **RoboSemanticBench** (**RSB**). (a) RSB turns a multiple-choice semantic question into an embodied answer-selection task: the VLA must understand the instruction, identify the correct option, bind it to the corresponding visible target, and execute the grasp. (b) The benchmark covers three semantic domains, RSB-Math, RSB-HardMath, and RSB-General, under both 4-choice and 10-choice settings. (c) Representative VLA models achieve low Task Success Rate (TSR) and often stay near or below the random-selection baseline, even though the manipulation primitive is simple. This highlights the semantic grounding failure in action prediction exposed by RSB.

follow multi-sentence compositional problem semantics before acting.

(iii) RSB-General covers non-mathematical semantic understanding, including commonsense QA about everyday tools, locations, household functions, and MMLU-derived multiple-choice questions (Hendrycks et al., 2021). Together, the three subsets probe semantic grounding in action prediction across controlled calculation, word-problem understanding, and general knowledge.

These sources are not meant to exceed modern language backbones: Qwen3-4B reports over 85% accuracy on GSM8K and over 70% on MMLU (Yang et al., 2025). Thus, RSB tests whether semantic competence expected from pretrained backbones is grounded in action prediction after robot fine-tuning.

3.3 Choice Suites and Visual Grounding

Each subset is instantiated in four-choice and ten-choice suites. The four-choice suite uses $\{A, B, C, D\}$; the ten-choice suite uses $\{A, B, C, D, E, F, G, H, I, K\}$, skipping J to avoid visual ambiguity with I. The larger suite expands the semantic action space and reduces success by guessing.

The ten-choice suite uses same-color letter blocks with black procedural strokes, making identity visible while reducing color shortcuts. Each episode randomizes block layout and option-to-letter assignment, so the target is determined jointly by the question, options, and mapping rather than a fixed object or position.

3.4 Instruction Generation and Leakage Control

Instructions are generated from templates that expose the question and option mappings but never the correct answer. Four-choice templates use placeholders such as $\{Q\}$ and $\{\text{MAPA}\}, \ldots, \{\text{MAPD}\}$, while ten-choice templates concatenate all mappings into a compact $\{\text{OPTIONS}\}$ field. Seen and unseen template variants test dependence on narrow surface forms.

Thus, the policy must interpret the question, identify the correct option value, and map that option to the visible block. Correct answers are stored only as evaluation metadata, preventing label leakage while enabling post-hoc diagnosis of semantic versus physical failures.

3.5 Expert Demonstrations and Simulation Evaluation

We instantiate RSB in a physics-based tabletop simulator with an Aloha-AgileX dual-arm embodiment, multi-view RGB cameras, wrist cameras, robot proprioception. The simulator randomizes candidate-block layout and option-to-block map-

ping while keeping the answer-selection primitive fixed, so the target is determined by the instruction rather than by position or appearance.

For demonstrations, a scripted expert maps the ground-truth answer to the visible block, chooses an arm by target position, and executes a motion-planned pick-and-place trajectory. Cartesian targets are converted into joint-space trajectories using MPLib with motion planning. Successful seeds are replayed to record trajectories.

At evaluation time, the policy receives only the scene observation and generated instruction, then predicts actions until termination or the step limit. The benchmark logs task success, grasp success, and semantic metadata for diagnosis; the exact metric definitions are given in Section 4.3. This logging is essential because RSB distinguishes two qualitatively different failures. If a policy never grasps a candidate block, the failure is mainly low-level control. If it grasps a candidate block but not the correct one, the policy has learned the motor primitive but not the semantic target selection. The latter case is the central semantic grounding failure studied in this paper.

4 Experiments

4.1 Experimental Protocol

All models follow the same train–test protocol. For each RSB suite in Table 3, we collect expert demonstrations from the training split, fine-tune the policy, and evaluate it on held-out semantic questions. Training and evaluation questions are disjoint to prevent memorizing question-answer pairs.

The number of training questions is determined by the source and cost of each semantic subset. RSB-Math uses 500 procedurally generated arithmetic questions, which already cover the controlled operator and distractor patterns while keeping expert trajectory generation lightweight. RSB-HardMath uses the full 7,473-question GSM8K training split, and RSB-General uses 10,000 sampled MMLU-style questions to provide broad commonsense and factual coverage. For each subset, the same question pool is used for both the 4-choice and 10-choice suites so that choice-set size changes while the underlying semantic distribution remains comparable.

To make comparison fair, we use comparable robot fine-tuning budgets, measured by training steps times batch size, whenever supported. Each model is reproduced and fine-tuned with its official codebase, using repository defaults unless otherwise stated.

At evaluation time, the policy receives only the observation and generated instruction; answer labels are never provided. For each model and suite, we run 500 simulation episodes and report average success rates. We release the full training data and simulation evaluation code for every RSB suite.

4.2 Evaluated Models

We select evaluated models based on representativeness and reproducibility. Specifically, we evaluate recent VLA models including GO1 (Shi et al., 2025), OpenVLA (Kim et al., 2024), DexVLA (Wen et al., 2025a), TinyVLA (Wen et al., 2025b), PD-VLA (Song et al., 2025), π_0 (Black et al., 2024), $\pi_{0.5}$ (Intelligence et al., 2025), GR00T N1.7 (NVIDIA et al., 2025), and QwenGR00T (Community, 2026). Table 1 reports fine-tuning configurations and results; Appendix F gives model-specific details.

4.3 Metrics

We report three metrics for each evaluation suite. **Task Success Rate (TSR)** measures the fraction of episodes in which the policy grasps the correct answer block specified by the semantic question and option mapping. **Grasp Success Rate (GSR)** measures the fraction of episodes in which the policy grasps any candidate answer block, regardless of correctness. Appendix C provides the detailed criteria for grasp success. To factor out differences in low-level grasping ability, we further define a **normalized Semantic Grounding score (nSG)**:

$$\mathrm{nSG} = \frac{\mathrm{TSR}/\mathrm{GSR} - 1/N}{1 - 1/N}, \tag{1}$$

where N is the number of candidate choices. This score measures whether a model selects the semantically correct target conditioned on successfully grasping a candidate block: $\mathrm{nSG} = 0$ corresponds to random target selection, while $\mathrm{nSG} = 1$ corresponds to perfect semantic target selection among successful grasps.

4.4 Main Results

Table 1 reports TSR across all suites, and Table 2 reports nSG after normalizing for grasp success. If a policy learns the manipulation primitive without reliable semantic understanding, TSR should remain low and nSG should stay near or below zero,

Model	Steps	Batch size	RSB-Math-4	RSB-Math-10	RSB-HardMath-4	RSB-HardMath-10	RSB-General-4	RSB-General-10	**Avg**
OpenVLA-OFT	100,000	64	10.3	3.5	20.5	7.6	16.7	8.0	**11.1**
GO1	100,000	16	3.8	2.2	1.4	2.0	0.2	2.4	**2.0**
DexVLA	100,000	64	13.6	2.1	5.3	1.8	12.2	3.7	**6.5**
TinyVLA	100,000	64	7.9	4.3	11.9	5.7	14.8	6.8	**8.6**
PD-VLA	100,000	64	10.9	5.3	9.8	5.1	14.2	8.8	**9.0**
π_0	100,000	64	13	5.4	25.8	6.6	18.0	7.6	**12.7**
$\pi_{0.5}$	100,000	64	32.8	12.0	25.8	16.2	24.2	19.6	**21.8**
GR00T N1.7	100,000	64	13.8	3.4	18.4	9.2	23.4	7.4	**12.6**
QwenGR00T	100,000	64	18.4	3.4	24.4	1.8	15.8	0.6	**10.7**

Table 1: Main evaluation results in **Task Success Rate (TSR, %)** after fine-tuning on expert demonstrations. Avg is computed over the six evaluation suites when all six are available. Full GSR/TSR decomposition is provided in Table 6 in the appendix.

Model	RSB-Math-4	RSB-Math-10	RSB-HardMath-4	RSB-HardMath-10	RSB-General-4	RSB-General-10	**Avg**
OpenVLA-OFT	-19.2	-6.9	-6.0	-0.9	-7.9	-2.2	**-7.2**
GO1	-27.8	-8.6	-30.1	-8.8	-33.0	-8.4	**-19.4**
DexVLA	-4.8	2.2	-5.1	-2.3	-14.6	4.3	**-3.4**
TinyVLA	-4.4	0.1	0.6	-0.7	1.7	4.1	**0.2**
PD-VLA	1.2	2.2	-2.1	4.3	14.0	-0.5	**3.2**
π_0	-14.3	-5.1	1.1	-3.8	-9.2	-2.7	**-5.7**
$\pi_{0.5}$	11.3	2.2	1.1	6.9	-1.1	10.7	**5.2**
GR00T N1.7	-14.3	-7.2	-8.6	-0.7	-1.6	-2.8	**-5.9**
QwenGR00T	-8.0	-7.2	-0.7	-9.1	-7.4	-10.4	**-7.1**

Table 2: **Normalized Semantic Grounding score (nSG, %)** for evaluated VLA models. nSG factors out grasp success by measuring semantic target selection conditioned on grasping a candidate block; 0 corresponds to random target selection and negative values indicate worse-than-random selection. Avg is computed over the six evaluation suites when all six are available.

Subset	Source	Choices	Train Questions
RSB-Math-4	easy arithmetic	4	500
RSB-Math-10	easy arithmetic	10	500
RSB-HardMath-4	GSM8K	4	7,473
RSB-HardMath-10	GSM8K	10	7,473
RSB-General-4	MMLU-style QA	4	10,000
RSB-General-10	MMLU-style QA	10	10,000

Table 3: Training-set statistics for each evaluation suite. Training and evaluation questions are disjoint within each subset.

especially in harder semantic domains and larger choice sets.

Appendix Table 6 provides the full GSR–TSR decomposition. High GSR with low TSR means the policy can grasp answer blocks but fails to select the semantically correct one; nSG asks whether this target selection is better than random among successful grasps.

Overall, most VLA models behave close to random target selection once low-level grasping is factored out: 25% for four-choice suites and 10% for ten-choice suites. Most average nSG scores are near or below zero, showing that successful grasps are not consistently guided by the semantic answer. This is not simply a failure to move the robot: several models achieve high GSR in the full decomposition, but their TSR remains low because they often grasp the wrong candidate. The gap is especially visible in the 10-choice suites, where the action space contains more plausible targets and shortcut-based selection becomes less reliable.

The main exception is $\pi_{0.5}$, which achieves the highest average TSR and the only clearly positive average nSG. A plausible explanation is that $\pi_{0.5}$ uses subtask annotations during robot-data pretraining, which may provide weak supervision for decomposing high-level instructions into intermediate semantic decisions and then following those decisions during action generation. Even so, its nSG remains modest, indicating that current VLA training is far from robust semantic grounding in action prediction.

4.5 Beyond Blocks: Everyday Object Targets

Figure 3: Examples of the *Beyond Blocks* setting, where candidate answer targets are replaced with everyday objects instead of uniform lettered blocks.

The main suites use lettered blocks to isolate semantic target selection from object-specific grasping difficulty. To test whether this proxy causes

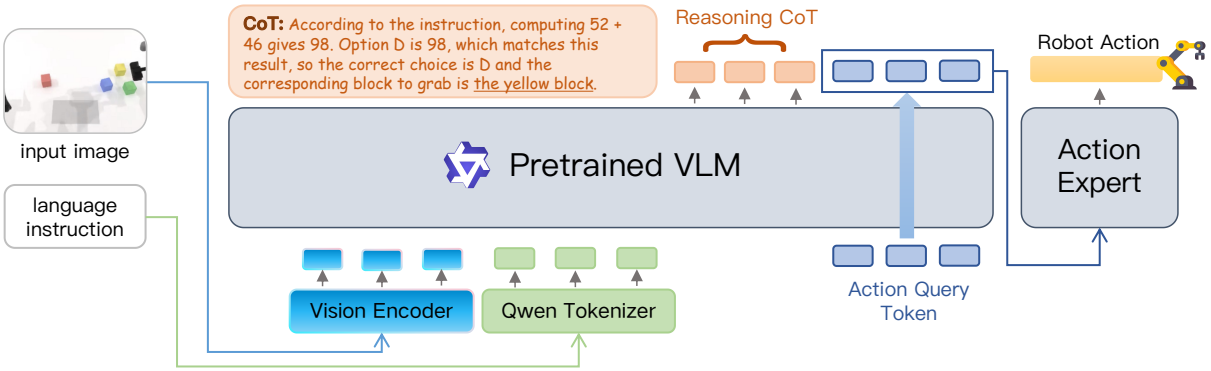

Figure 4: Overview of **ReasoningVLA**. The VLM generates a textual CoT, then uses Action Query Tokens to pass semantic context to the Action Expert for action-chunk generation.

the observed failures, we replace candidate blocks with everyday objects such as toy cars, playing cards, and shoes, and build three scenes matching the three RSB semantic levels. Using the same GSR/TSR protocol, Appendix B shows that grasp success remains high while TSR stays low. Thus, the benchmark difficulty mainly comes from semantic grounding in action prediction rather than from the block interface, making blocks a controlled and sufficient proxy.

5 Failed Exploration

Beyond benchmarking existing VLA models, we explored two natural interventions for reducing the semantic grounding gap. One makes the VLA verbalize an intermediate semantic solution before predicting actions; the other adds language-centric supervision during robot fine-tuning. Both are negative results: neither reliably makes pretrained semantic understanding participate in action prediction.

5.1 Exploration 1: ReasoningVLA

ReasoningVLA makes the semantic decision explicit before action generation. It trains the model to produce a textual CoT identifying the target option block, then conditions action prediction on this intermediate solution. This tests a natural hypothesis: if the VLM backbone can first solve the semantic question in language space, the downstream action module may have an easier target-selection problem. The risk is that the generated answer must still be bound to the current scene and preserved through continuous action generation.

Architecture. ReasoningVLA uses a Qwen3-VL-4B backbone and a DiT-based Action Expert, as shown in Figure 4. The VLM autoregressively generates a CoT between `<think>` and `</think>`, then appends eight `<|action|>` Action Query Tokens. Their last-layer hidden states are passed to the Action Expert through cross-attention, and the Action Expert generates an action chunk with flow matching. The Action Query Tokens therefore serve as the interface between the textual semantic solution and the continuous action generator.

CoT supervision is distilled from Gemini 3 Flash (Gemini Team, 2025) and attached to each robot demonstration; Appendix D provides the prompt format and examples.

Training Objective. ReasoningVLA optimizes a weighted action-generation and CoT-generation objective:

$$\mathcal{L} = 0.9\,\mathcal{L}_{\mathrm{FM}} + 0.1\,\mathcal{L}_{\mathrm{CoT}}. \tag{2}$$

Here $\mathcal{L}_{\mathrm{FM}}$ supervises the action chunk and $\mathcal{L}_{\mathrm{CoT}}$ supervises next-token prediction for the distilled CoT.

Table 4 shows that ReasoningVLA improves average TSR over QwenGR00T, but its GSR decreases and the absolute TSR remains low, especially in 10-choice suites. This suggests that CoT supervision can help recover part of the semantic target selection ability, but it does not reliably ground the selected answer into robot actions. Explicit reasoning traces alone are therefore insufficient for semantic grounding in action prediction.

5.2 Exploration 2: VLA Cotrain

The second attempt is to preserve the VLM backbone's semantic competence by adding VQA supervision during robot fine-tuning. As shown in Table 4, this cotraining strategy does not improve semantic grounding in action prediction: TSR drops on all six RSB suites, and the average TSR decreases from 10.7% to 8.2%. This suggests that

Method	RSB-Math-4	RSB-Math-10	RSB-HardMath-4	RSB-HardMath-10	RSB-General-4	RSB-General-10	TSR Avg
QwenGR00T (baseline)	18.4	3.4	24.4	1.8	15.8	0.6	10.7
ReasoningVLA	27.0 ↑	7.3 ↑	28.6 ↑	5.8 ↑	20.6 ↑	6.8 ↑	16.0 ↑
QwenGR00T (Cotrain)	13.2 ↓	2.4 ↓	20.2 ↓	0.8 ↓	12.2 ↓	0.2 ↓	8.2 ↓

Table 4: TSR results for failed exploration attempts compared with the QwenGR00T baseline. ReasoningVLA improves TSR but remains far from reliable semantic grounding in action prediction, while VLA cotraining consistently reduces TSR. TSR Avg is the mean over the six evaluation suites.

language-centric auxiliary supervision may conflict with the action-learning objective and perturb representations needed for VLA adaptation, consistent with the finding of VLM4VLA (Zhang et al., 2026a). Appendix E provides the training details.

6 Error Analysis

We analyze RSB-Math episodes where the policy grasps a candidate block but still fails the task. Since the robot has interacted with an answer block, these errors directly reveal whether the selected target follows the instruction semantics rather than whether the robot can execute a grasp.

Error type	ReasoningVLA	QwenGR00T
Grasped but not placed correctly	4.36	4.08
Incorrect CoT, wrong target	6.70	–
Correct CoT, wrong target	89.93	–
Wrong target after grasp	–	95.92

Table 5: Error analysis on RSB-Math for episodes with grasp success but task failure. Values are percentages within this failure subset.

Table 5 shows that most failures are target-selection errors rather than placement failures. QwenGR00T chooses the wrong target in 95.92% of grasp-success/task-failure cases, and ReasoningVLA shows the same pattern. This supports the central diagnosis of RSB: current VLA models can often learn the answer-block manipulation primitive, but they still do not know which block should be selected from the instruction.

ReasoningVLA further exposes a *reasoning-following* failure. Only 6.70% of its grasp-success/task-failure episodes are associated with an incorrect CoT; most occur when the CoT identifies the correct answer but the action still grasps the wrong block. Thus, improving language-space semantic decisions is not sufficient unless the action pathway reliably follows those decisions.

7 Discussion

What does RSB diagnose? RSB evaluates semantic grounding in action prediction rather than standalone question answering. It asks whether a VLA model can use instruction semantics to choose the correct physical target during action prediction, after accounting for its ability to grasp candidate objects. This distinction is important: a VLA may contain a capable pretrained backbone, yet still act as if it does not understand the instruction if the action pathway relies on imitation shortcuts, visual priors, or poorly routed semantic features. The GSR–TSR decomposition and nSG score make this gap visible by separating grasping ability from semantic target selection.

Implications for VLA training. The results suggest that simply attaching a strong VLM to an action expert is not enough to obtain semantically grounded action prediction. A more promising direction may require training objectives and interfaces that explicitly preserve the selected semantic target and expose it to the action module in a stable, scene-grounded form. In this sense, RSB provides not only an evaluation suite, but also a diagnostic target for future VLA architectures: successful models should maintain high grasp success while raising TSR and nSG far above random selection.

8 Conclusion

We presented RoboSemanticBench (RSB), a benchmark for diagnosing semantic grounding in VLA action prediction. By converting math, hard-math, and general-knowledge questions into embodied answer-selection tasks, RSB makes the correct action depend on instruction understanding rather than visual or action-distribution shortcuts. Across representative VLA models, many policies learn the grasping primitive, but their target choices remain near or below random once grasp success is controlled for. This reveals that current VLA training often fails to route semantic decisions from the pretrained backbone into the action pathway, and motivates future VLA systems whose actions are genuinely grounded in instruction semantics.

Limitations

RoboSemanticBench is a simulation benchmark and focuses on tabletop answer selection rather than full long-horizon household manipulation. Although the Beyond Blocks setting suggests that the block interface is a reasonable proxy for semantic grounding in action prediction, real-world deployment may introduce additional perception, contact, and safety challenges. The benchmark also evaluates multiple-choice semantic grounding, so it does not cover open-ended dialogue, clarification, or interactive task repair.

Ethics Statement

This work introduces a simulation benchmark for evaluating semantic grounding in action prediction for VLA models. It does not involve human subjects, private user data, or real-world robot deployment. The semantic questions are derived from public or synthetic question sources, and the robot evaluations are conducted in controlled tabletop simulation. The main ethical risk is misinterpretation: high benchmark scores should not be treated as evidence that a VLA model is safe or reliable for open-world physical deployment. Conversely, low scores should be understood as a diagnostic signal about semantic grounding under the specific embodied answer-selection setting, rather than as a complete characterization of a model's capabilities. We encourage using RoboSemanticBench as a research diagnostic for improving language-grounded robot learning, together with additional safety, robustness, and real-world validation before any deployment in physical environments.

References

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, and 1 others. 2024. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, and 35 others. 2023. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *Preprint*, arXiv:2307.15818.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, and 1 others. 2025. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*.

Suhwan Choi, Yunsung Lee, Yubeen Park, Chris Dongjoo Kim, Ranjay Krishna, Dieter Fox, and Youngjae Yu. 2026. vla-eval: A unified evaluation harness for vision-language-action models. *Preprint*, arXiv:2603.13966.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

StarVLA Community. 2026. Starvla: A lego-like codebase for vision-language-action model developing. *arXiv preprint arXiv:2604.05014*.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, and 3 others. 2023. Palm-e: An embodied multimodal language model. *Preprint*, arXiv:2303.03378.

Irving Fang, Juexiao Zhang, Shengbang Tong, and Chen Feng. 2025. From intention to execution: Probing the generalization boundaries of vision-language-action models. *Preprint*, arXiv:2506.09930.

Yu Fang, Yuchun Feng, Dong Jing, Jiaqi Liu, Yue Yang, Zhenyu Wei, Daniel Szafir, and Mingyu Ding. 2026. When vision overrides language: Evaluating and mitigating counterfactual failures in vlas. *Preprint*, arXiv:2602.17659.

Gemini Team. 2025. A new era of intelligence with gemini 3.

Xinyu Guo, Bin Xie, Wei Chai, Xianchi Deng, Tiancai Wang, Zhengxing Wu, and Xingyu Chen. 2026. Priorvla: Prior-preserving adaptation for vision-language-action models. *Preprint*, arXiv:2605.10925.

Asher J. Hancock, Xindi Wu, Lihan Zha, Olga Russakovsky, and Anirudha Majumdar. 2025. Actions as language: Fine-tuning vlms into vlas without catastrophic forgetting. *Preprint*, arXiv:2509.22195.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Yuchen Hou and Lin Zhao. 2026. Langgap: Diagnosing and closing the language gap in vision-language-action models. *Preprint*, arXiv:2603.00592.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, and 17 others. 2025. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *Preprint*, arXiv:2504.16054.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. 2024. Openvla: An open-source vision-language-action model. In *Annual Conference on Robot Learning (CoRL)*.

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Oier Mees, Karl Pertsch, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, Sergey Levine, Jiajun Wu, Chelsea Finn, Hao Su, Quan Vuong, and Ted Xiao. 2024. Evaluating real-world robot manipulation policies in simulation. In *Annual Conference on Robot Learning (CoRL)*.

Shijie Lian, Bin Yu, Xiaopeng Lin, Laurence T. Yang, Zhaolong Shen, Changti Wu, Yuzhuo Miao, Cong Huang, and Kai Chen. 2026. Langforce: Bayesian decomposition of vision language action models via latent action queries. *Preprint*, arXiv:2601.15197.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. 2023. Libero: Benchmarking knowledge transfer for lifelong robot learning. *arXiv preprint arXiv:2306.03310*.

Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. 2022. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters (RA-L)*, 7(3):7327–7334.

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

NVIDIA, :, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, and 24 others. 2025. Gr00t n1: An open foundation model for generalist humanoid robots. *Preprint*, arXiv:2503.14734.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, and 1 others. 2024. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE.

Pierre Sermanet, Tianli Ding, Jeffrey Zhao, Fei Xia, Debidatta Dwibedi, Keerthana Gopalakrishnan, Christine Chan, Gabriel Dulac-Arnold, Sharath Maddineni, Nikhil J Joshi, Pete Florence, Wei Han, Robert Baruch, Yao Lu, Suvir Mirchandani, Peng Xu, Pannag Sanketi, Karol Hausman, Izhak Shafran, and 2 others. 2023. Robovqa: Multimodal long-horizon reasoning for robotics. In *arXiv preprint arXiv:2311.00899*.

Modi Shi, Yuxiang Lu, Huijie Wang, and Shaoze Yang. 2025. Open-sourcing go-1: The bitter lessons of building vla systems at scale. https://opendrivelab.com/OpenGO1/. Blog post.

Wenxuan Song, Jiayi Chen, Pengxiang Ding, Han Zhao, Wei Zhao, Zhide Zhong, Zongyuan Ge, Jun Ma, and Haoang Li. 2025. Accelerating vision-language-action model integrated with action chunking via parallel decoding. *arXiv preprint arXiv:2503.02310*.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. 2024. Octo: An open-source generalist robot policy. *Preprint*, arXiv:2405.12213.

Selma Wanna, Agnes Luhtaru, Jonathan Salfity, Ryan Barron, Juston Moore, Cynthia Matuszek, and Mitch Pryor. 2026. Limited linguistic diversity in embodied ai datasets. *Preprint*, arXiv:2601.03136.

Junjie Wen, Yichen Zhu, Jinming Li, Zhibin Tang, Chaomin Shen, and Feifei Feng. 2025a. Dexvla: Vision-language model with plug-in diffusion expert for general robot control. *arXiv preprint arXiv:2502.05855*.

Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Kun Wu, Zhiyuan Xu, Ning Liu, Ran Cheng, Chaomin Shen, Yaxin Peng, and 1 others. 2025b. Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation. In *IEEE Robotics and Automation Letters (RA-L)*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Bin Yu, Shijie Lian, Xiaopeng Lin, Yuliang Wei, Zhaolong Shen, Changti Wu, Yuzhuo Miao, Xinming Wang, Bailing Wang, Cong Huang, and

Kai Chen. 2026. Twinbrainvla: Unleashing the potential of generalist vlms for embodied tasks via asymmetric mixture-of-transformers. *Preprint*, arXiv:2601.14133.

Borong Zhang, Jiahao Li, Jiachen Shen, Yishuai Cai, Yuhao Zhang, Yuanpei Chen, Juntao Dai, Jiaming Ji, and Yaodong Yang. 2025a. Vla-arena: An open-source framework for benchmarking vision-language-action models. *Preprint*, arXiv:2512.22539.

Chuheng Zhang, Rushuai Yang, Xiaoyu Chen, Kaixin Wang, Li Zhao, Yi Chen, and Jiang Bian. 2025b. How do vlas effectively inherit from vlms? *Preprint*, arXiv:2511.06619.

Jianke Zhang, Xiaoyu Chen, Qiuyue Wang, Mingsheng Li, Yanjiang Guo, Yucheng Hu, Jiajun Zhang, Shuai Bai, Junyang Lin, and Jianyu Chen. 2026a. Vlm4vla: Revisiting vision-language-models in vision-language-action models. *Preprint*, arXiv:2601.03309.

Jianke Zhang, Yuanfei Luo, Yucheng Hu, Xiaoyu Chen, Yanjiang Guo, Ziyang Liu, Hongbin Xu, Tian Lan, and Jianyu Chen. 2026b. Uam: A dual-stream perspective on forgetting in vla training. *Preprint*, arXiv:2605.15735.

Ninghao Zhang, Bin Zhu, Shijie Zhou, and Jingjing Chen. 2026c. Restoring linguistic grounding in vla models via train-free attention recalibration. *Preprint*, arXiv:2603.06001.

Shiduo Zhang, Zhe Xu, Peiju Liu, Xiaopeng Yu, Yuan Li, Qinghui Gao, Zhaoye Fei, Zhangyue Yin, Zuxuan Wu, Yu-Gang Jiang, and Xipeng Qiu. 2024. Vlabench: A large-scale benchmark for language-conditioned robotics manipulation with long-horizon reasoning tasks. *Preprint*, arXiv:2412.18194.

Zhongyi Zhou, Yichen Zhu, Minjie Zhu, Junjie Wen, Ning Liu, Zhiyuan Xu, Weibin Meng, Ran Cheng, Yaxin Peng, Chaomin Shen, and Feifei Feng. 2025. Chatvla: Unified multimodal understanding and robot control with vision-language-action model. In *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

A Full Main Evaluation Results

Table 6 reports the full GSR/TSR decomposition for all evaluated models. The Avg columns summarize the mean GSR and TSR over the six RSB evaluation suites when all six results are available.

B Beyond Blocks Results

Tables 7 and 8 report the Beyond Blocks results for $\pi_{0.5}$ and GR00T N1.7. After replacing lettered blocks with everyday objects, both models show evaluation trends consistent with the block-based setting: GSR remains high, while TSR remains much lower than grasp success. This indicates that the main bottleneck is still semantic target selection rather than object-specific grasping difficulty. Therefore, the block-based RSB task is sufficient as a controlled proxy for evaluating semantic grounding in current VLA action prediction.

C Grasp Success Criteria

Grasp success detection. GSR measures the percentage of episodes in which the policy successfully grasps any candidate answer block, regardless of whether the grasped block corresponds to the correct answer. A grasp is counted only when all of the following conditions are satisfied: (1) one gripper is in contact with a candidate block; (2) at least one of the left or right grippers is closed; and (3) the contacted block is either lifted above the tabletop or remains in stable gripper contact for at least eight consecutive simulation steps. This definition avoids counting accidental gripper closure, empty grasps, or contacts with non-candidate objects as grasp success. A policy that grasps the wrong candidate block is therefore counted as successful under GSR but unsuccessful under TSR, which is what allows the GSR–TSR gap to diagnose semantic target-selection failures.

Normalized semantic grounding score. The nSG score is computed only when GSR is non-zero and should be interpreted together with GSR. It measures semantic target selection conditioned on the policy having successfully grasped a candidate block. Because the score subtracts the random-selection baseline $1/N$, values near zero indicate random semantic selection among grasped candidates, positive values indicate better-than-random semantic grounding, and negative values indicate worse-than-random target selection.

D ReasoningVLA Data Construction

CoT annotation source. Default VLA demonstrations contain observations, language instructions, and expert action chunks, but they do not include textual explanations for the semantic decision behind each action. We therefore augment the RSB training demonstrations with CoT annotations distilled from Gemini 3 Flash (Gemini Team, 2025). For each training instruction, the teacher model receives the multiple-choice question and options, solves the semantic problem, and outputs a short rationale that includes the expression, com-

Model	Steps	Batch size	RSB-Math-4		RSB-Math-10		RSB-HardMath-4		RSB-HardMath-10		RSB-General-4		RSB-General-10		**Avg**	
			GSR	TSR	GSR	TSR	GSR	TSR	GSR	TSR	GSR	TSR	GSR	TSR	GSR	TSR
OpenVLA-OFT	100,000	64	97	10.3	92.3	3.5	100.0	20.5	83.1	7.6	87.5	16.7	100.0	8.0	**93.3**	**11.1**
GO1	100,000	16	91.2	3.8	96.0	2.2	58.4	1.4	96.8	2.0	70.8	0.2	100.0	2.4	**85.5**	**2.0**
DexVLA	100,000	64	63.6	13.6	17.5	2.1	25.0	5.3	22.7	1.8	87.0	12.2	26.7	3.7	**40.4**	**6.5**
TinyVLA	100,000	64	36.4	7.9	42.8	4.3	46.8	11.9	61.0	5.7	56.3	14.8	49.6	6.8	**48.8**	**8.6**
PD-VLA	100,000	64	42.1	10.9	44.2	5.3	41.8	9.8	36.8	5.1	40.0	14.2	92.3	8.8	**49.5**	**9.0**
π_0	100,000	64	91.0	13.0	100.0	5.4	100.0	25.8	100.0	6.6	99.4	18.0	100.0	7.6	**98.4**	**12.7**
$\pi_{0.5}$	100,000	64	98.0	32.8	100.0	12.0	100.0	25.8	100	16.2	100.0	24.2	100.0	19.6	**99.7**	**21.8**
GR00T N1.7	100,000	64	96.6	13.8	97.6	3.4	99.2	18.4	98.0	9.2	98.2	23.4	99.2	7.4	**98.1**	**12.6**
QwenGR00T	100,000	64	96.8	18.4	97.6	3.4	99.6	24.4	98.8	1.8	81.2	15.8	96.8	0.6	**95.1**	**10.7**

Table 6: Full GSR/TSR decomposition for the main evaluation. GSR measures grasping any candidate block, while TSR requires grasping the correct answer block. Avg reports the mean GSR and TSR over all six evaluation suites when all six are available. All success rates are percentages and higher is better.

Scene	GSR	TSR
RSB-Math-4	96.2	32.4
RSB-Math-10	96.8	11.8
RSB-HardMath-4	97.4	24.6
RSB-HardMath-10	96.0	15.8
RSB-General-4	97.4	23.6
RSB-General-10	97.2	19.2
Avg	**96.8**	**21.2**

Table 7: Beyond Blocks evaluation results for $\pi_{0.5}$. GSR measures grasping any candidate object, while TSR requires grasping the object associated with the correct semantic answer.

Scene	GSR	TSR
RSB-Math-4	95.2	13.4
RSB-Math-10	96.4	3.2
RSB-HardMath-4	98.4	18.8
RSB-HardMath-10	97.8	9.4
RSB-General-4	96.8	22.6
RSB-General-10	97.2	7.0
Avg	**97.0**	**12.4**

Table 8: Beyond Blocks evaluation results for GR00T N1.7. GSR measures grasping any candidate object, while TSR requires grasping the object associated with the correct semantic answer.

puted answer, matched option, and corresponding color block.

Prompt format. We use a fixed system prompt to constrain the teacher to produce concise annotations and to follow a deterministic option-to-color mapping:

System Prompt

```
You are a data annotation assistant for robot block-picking tasks. Given an instruction, generate a concise solution rationale. Only output one paragraph. Do not output JSON or extra explanations. The reasoning must include the math expression, computed result, matched option, and corresponding color block. The fixed color mapping is: A=red block, B=green block, C=blue block, D=yellow block.
```

The resulting rationale is then wrapped with <think> and </think> before being used as CoT supervision. We discard teacher responses that do not contain a unique final option or that are inconsistent with the ground-truth answer metadata.

Example annotation. For an instruction such as "what is 27 minus 17? options: (A) 4 (B) 10 (C) 11 (D) 14. place the correct option block in the gray answer zone," the distilled annotation is:

Example CoT Annotation

<think> 27 minus 17 equals 10. Among the options, option B is 10. According to the fixed color mapping, option B corresponds to the green block, so the robot should grasp the green block. </think>

This CoT is concatenated with the original observation–instruction demonstration. The action target remains the same expert action chunk, so the augmented sample supervises both the language-space semantic solution and the continuous action-generation pathway.

E VLA Cotraining Details

Motivation. The cotraining experiment tests whether language-centric supervision can preserve the VLM backbone's semantic competence during robot fine-tuning. The baseline QwenGR00T is trained only on RSB robot demonstrations, where each sample contains observations, a semantic instruction, and an expert action chunk. The cotraining variant keeps the same robot demonstration data and evaluation protocol, but additionally injects visual question answering (VQA) samples from RoboVQA (Sermanet et al., 2023) during fine-tuning so that the shared VLM backbone is optimized for both semantic question answering

and action prediction.

Training mixture. We implement cotraining based on the starVLA framework. Mixed fine-tuning batches are constructed from two sources: robot-demonstration samples use the standard VLA input format and supervise the action-generation pathway with expert action chunks, while VQA samples use image–question–answer pairs and supervise the language modeling pathway with next-token prediction over the answer text. The two data streams share the same VLM backbone, while the action expert is updated only by robot-demonstration samples. This design is intended to regularize the Semantic Expert without changing the robot task definition or giving the policy access to evaluation answers.

Training setup. We train the cotraining model for 100,000 steps on 8 NVIDIA H100 GPUs using DeepSpeed ZeRO-2 optimization. The global batch size is 64 for the VLA robot-demonstration stream and 32 for the VQA stream.

Objective. The cotraining objective combines the original VLA action loss with a language modeling loss on VQA answers:

$$\mathcal{L}_{\text{cotrain}} = \mathcal{L}_{\text{action}} + 0.1\,\mathcal{L}_{\text{VQA}}, \tag{3}$$

where $\mathcal{L}_{\text{action}}$ is the action-generation loss used by the QwenGR00T baseline and $\mathcal{L}_{\text{VQA}}$ is a next-token prediction loss for RoboVQA responses. The VQA branch therefore encourages the backbone to retain language-space semantic answering ability, while the action branch continues to optimize imitation learning on expert robot trajectories.

Evaluation. After cotraining, the model is evaluated exactly like the baseline: it receives only the simulator observation and the generated RSB instruction, and must output robot actions without access to the correct answer label or VQA supervision. This makes the comparison in Table 4 a direct test of whether preserving language-oriented supervision during fine-tuning improves semantic grounding in action prediction.

F Training Details for Evaluated Models

All evaluated models are trained on 8 NVIDIA H100 GPUs. To make the training budget comparable across models, we keep the total number of optimization samples fixed by matching *training steps* $\times$ *global batch size*. Other model-specific training details, including optimizer settings, learning-rate schedules, precision settings, and architecture-specific data formatting, follow the default configurations of the corresponding official codebases whenever possible.

GO1. GO1 is an open generalist VLA robotic foundation model released by OpenDriveLab, designed as a scalable policy for language-conditioned robot control (Shi et al., 2025). We reproduce GO1 with its official codebase and fine-tune it on RSB demonstrations using the observation, instruction, and action interface expected by the released implementation.

OpenVLA. OpenVLA is an open-source VLA model that adapts a pretrained vision-language backbone into an autoregressive robot policy, representing robot actions as tokens predicted from visual observations and language instructions (Kim et al., 2024). We fine-tune OpenVLA from its pretrained checkpoint on the RSB training split and evaluate the resulting policy with the same simulator protocol as the other models.

DexVLA. DexVLA augments a VLM backbone with a plug-in diffusion expert, using the VLM for semantic perception and instruction processing while delegating continuous robot control to a diffusion-based action module (Wen et al., 2025a). We fine-tune DexVLA on the same semantic answer-selection demonstrations used for the other evaluated models.

TinyVLA. TinyVLA is a compact and data-efficient VLA architecture designed to reduce inference cost while maintaining robot manipulation performance (Wen et al., 2025b). We train TinyVLA on the RSB expert demonstrations as a small VLA baseline under the same train–test split.

PD-VLA. PD-VLA targets VLA models with action chunking and accelerates autoregressive action decoding through parallel fixed-point decoding, preserving the underlying action-chunking policy interface while improving inference efficiency (Song et al., 2025). We fine-tune and evaluate the released PD-VLA-style implementation on the RSB training and evaluation suites.

π_0 and $\pi_{0.5}$. π_0 is a generalist VLA flow model built around a pretrained PaliGemma-style VLM backbone and an action expert, using a mixture-of-transformers style interface to connect semantic

processing with continuous action generation via flow matching (Black et al., 2024). $\pi_{0.5}$ extends this family toward open-world generalization and uses additional robot-data pretraining signals, including subtask-style supervision, that can help decompose instructions before action prediction (Intelligence et al., 2025). We fine-tune both models on the RSB expert demonstrations with their official training pipelines.

GR00T N1.7. GR00T N1 is an open foundation model for generalist humanoid robots, combining multimodal instruction understanding with robot action generation for whole-body or manipulation-oriented control (NVIDIA et al., 2025). We fine-tune GR00T N1.7 on the corresponding RSB training split and evaluate it using the same 500-episode simulator protocol.

QwenGR00T. QwenGR00T is a Qwen-backed VLA implementation built within the StarVLA codebase, which provides a modular framework for constructing VLA policies from interchangeable VLM backbones and action experts (Community, 2026). We fine-tune QwenGR00T on the RSB demonstrations using the same train–test split as the other VLA models.