PROGRESSLM: Towards Progress Reasoning in Vision-Language Models

Anonymous ACL submission

Abstract

Estimating task progress requires long-horizon and dynamic reasoning, going beyond static visual perception. Although Vision–Language Models (VLMs) excel at describing what is visible in a single observation, it remains unclear whether they can infer *how far a task has progressed* from partial information. To study this question, we introduce **PROGRESS-BENCH**, a benchmark with over 3K instances for evaluating progress reasoning from a single observation. We further examine a human-inspired two-stage paradigm that combines episodic retrieval with mental simulation. We instantiate this paradigm through both training-free prompting and a training-based approach using the automatically curated PROGRESSLM-45K dataset. Experiments on 14 VLMs show that most models struggle with reliable progress estimation, and that training-free reasoning provides only limited and model-dependent benefits. In contrast, the training-based PROGRESSLM-3B achieves consistent improvements in accuracy, robustness to viewpoint variation, and handling of unanswerable cases, despite its small scale. Additional analyses reveal common failure patterns in existing VLMs and clarify when and why progress reasoning succeeds or fails.

1 Introduction

Given an observation from an intermediate moment of a task, most Vision Language Models (VLMs) (Hurst et al., 2024; Bai et al., 2025; Wang et al., 2025b) can accurately describe what is visible. However, estimating *how much of the task has been completed* is fundamentally different, as it requires long-horizon, dynamic reasoning beyond snapshot-level perception.

Prior work on progress estimation either relies on task-specific regression models (Yang et al., 2024; Chen et al., 2025), or infers progress indirectly via surrogate objectives such as trajectory reordering (Ma et al., 2024b) or pairwise comparison (Zhai et al., 2025). This raises a fundamental question: *can VLMs acquire progress estimation as a general reasoning capability from a single observation?* To systematically study this problem, we introduce **PROGRESS-BENCH**, a benchmark built on the robotic manipulation domain (Wu et al., 2025b), where task execution follows clear and temporally ordered progressions. Each instance consists of a task demonstration and a single observation, and the model is required to predict a numerical progress score. The benchmark is designed to probe progress reasoning along three main axes: demonstration modality (vision vs. text), viewpoint correspondence (same-view vs. cross-view between the demonstration and the observation), and answerability.

Beyond benchmarking existing models, we further ask: **How, then, can progress reasoning be effectively learned?** Humans excel at progress estimation by interpreting task execution as a continuous process that combines *episodic retrieval* to locate a coarse anchor along the task trajectory, and *mental simulation* to reason about how the task state evolves from this anchor toward the current observation (Schacter et al., 2008). Inspired by this process, we first explore **training-free** prompting strategies that explicitly encourage VLMs to follow this two-stage reasoning pattern as shown in Figure 1. To further endow models with robust progress reasoning ability, we explore a **training-based** approach and automatically construct a dataset named **PROGRESSLM-45K**, with 25K chain-of-thought samples for supervised cold-start and 20K used for reinforcement learning refinement, yielding a progress-reasoning-

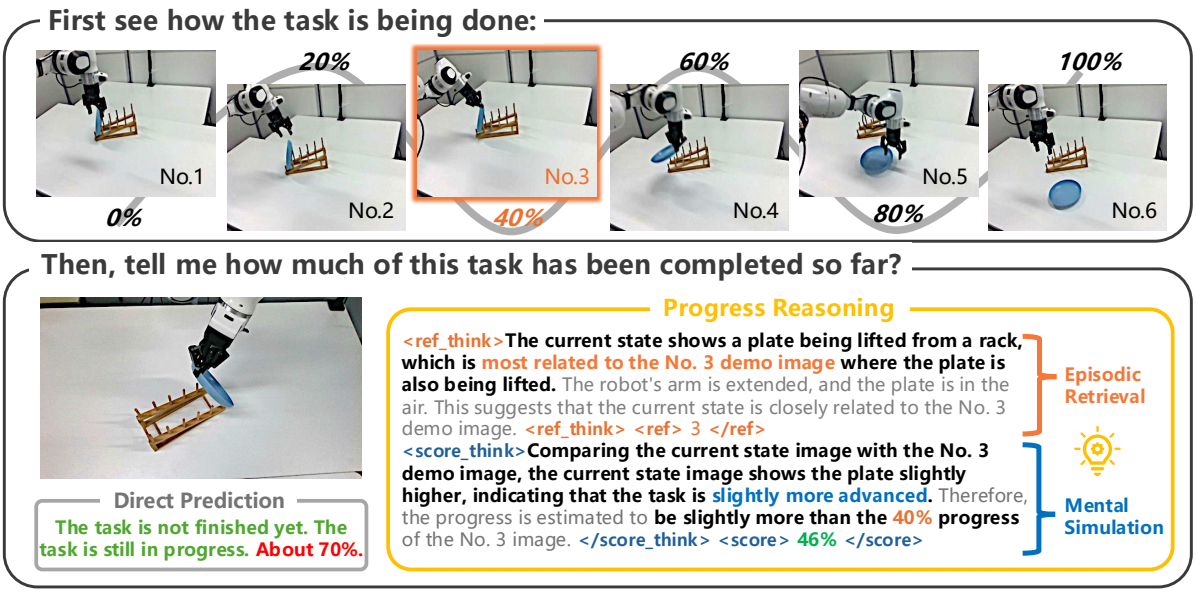


Figure 1: Given a task demonstration and a single observation, direct prediction can often judge whether the task is unfinished, but struggles to assign a well-calibrated progress score. Progress reasoning instead follows a coarse-to-fine process: it first performs *episodic retrieval* to coarsely locate the observation along the demonstrated task, then applies *mental simulation* to imagine the transition from the retrieved anchor to the current observation, yielding accurate and interpretable progress estimation.

enhanced model, **PROGRESSLM-3B**.

Our experiments across 14 models show that VLMs struggle to estimate task progress reliably from a single observation. Direct prediction leads to strong sensitivity to demonstration modality and viewpoint changes, as well as poor handling of unanswerable cases. Training-free progress reasoning provides only conditional benefits. In contrast, training-based PROGRESSLM-3B yields consistent improvements even at small model scale. Further analysis reveals that different models' error patterns. We additionally examine when progress reasoning can be effective and why demonstration modality plays a critical role.

Our main contributions are as follows:

- We introduce PROGRESS-BENCH, a benchmark with over 3K instances for systematically explore whether VLMs can perform progress reasoning from a single observation under variations of demonstration modality, viewpoint, and answerability.
- We benchmark 14 VLMs on PROGRESS-BENCH and show that existing models exhibit limited and unstable progress reasoning, with strong sensitivity to modality and viewpoint changes, poor handling of unanswerable cases, and frequent collapse to coarse or heuristic predictions.
- We investigate how progress reasoning can be improved through a human-inspired two-stage paradigm. While training-free prompting yields only conditional gains, explicit training leads to PROGRESSLM-3B, which achieves performance comparable to or surpassing GPT-5 on PROGRESS-BENCH. Further analyses reveal common failure patterns in existing VLMs and clarify when and why progress reasoning succeeds or fails.

2 PROGRESS-BENCH

Overview. PROGRESS-BENCH evaluates whether a model can infer task progress from a single observation by situating it within the temporal structure of an ongoing task, going beyond static perception toward progression reasoning. Each instance consists of a task demonstration D and an observation o sampled from an intermediate execution stage. The demonstration covers the full task from start to completion and is presented either as a sequence of images or as stepwise textual actions. Given (D, o), the model predicts a normalized progress score $p \in [0, 100\%]$ indicating how far the task has advanced.

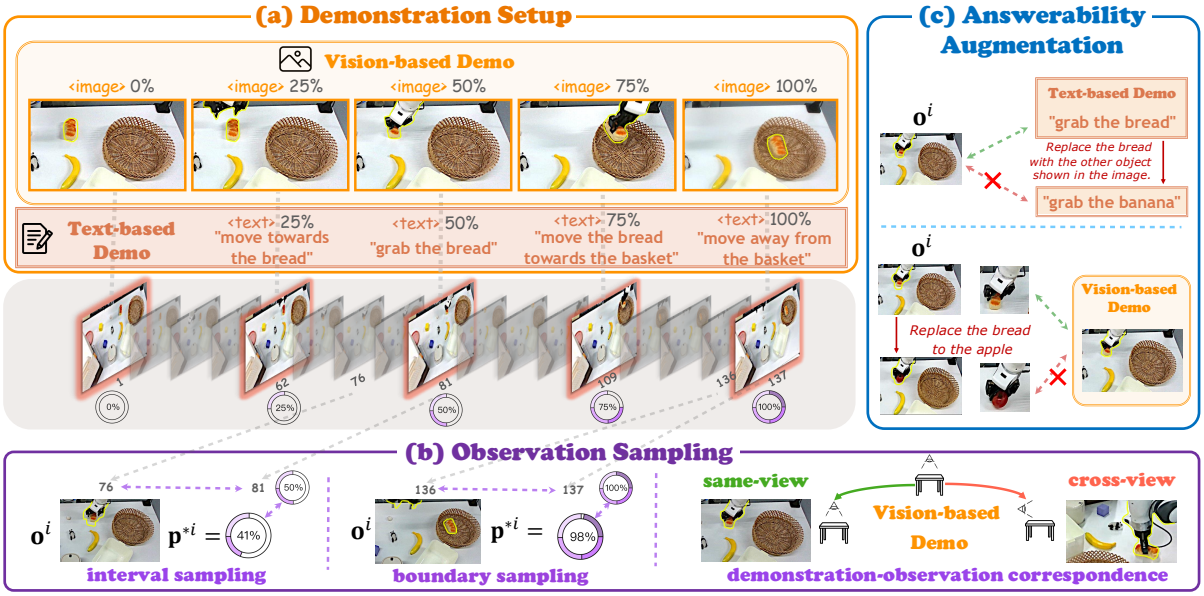


Figure 2: Overview of PROGRESS-BENCH. (a) **Demonstration setup** provides vision-based key-frame demonstrations or text-based step descriptions with progress annotations. (b) **Observation sampling** selects observations between or near demonstration steps, with progress assigned by interpolation; vision-based settings include same-view and cross-view cases. (c) **Answerability augmentation** creates unanswerable samples by introducing mismatches between demonstrations and observations.

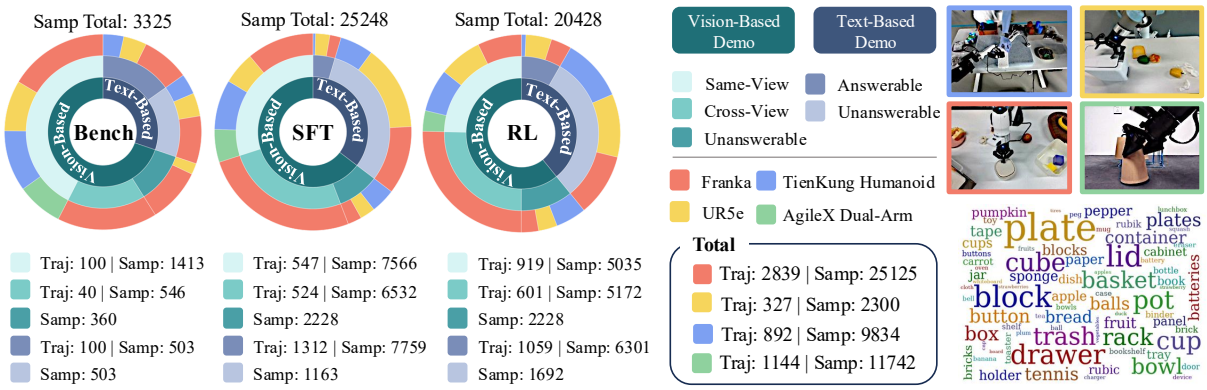


Figure 3: Data statistics of PROGRESS-BENCH and PROGRESSLM-45K (25K SFT, 20K RL). Traj and Samp indicate trajectory and sample counts; right panels show robot embodiments and object diversity.

When the demonstration is insufficient or inconsistent with the observation, the model should instead output N/A. To systematically probe this capability, PROGRESS-BENCH is constructed by varying demonstration modality, observation–demonstration viewpoint correspondence, and answerability, enabling controlled analysis of perception, temporal reasoning, and uncertainty handling.

2.1 Benchmark Construction

Demonstration Setup. We consider two demonstration modality types. *Vision-based* demonstrations consist of key frames from expert executions, $\mathcal{D}_v = \{(\mathbf{f}_{a_j}, \mathbf{p}_j)\}_{j=1}^{N}$, where each frame depicts a complete task state. *Text-based* demonstrations provide stepwise action descriptions, $\mathcal{D}_t = \{(\mathbf{t}_j, \mathbf{p}_j)\}_{j=1}^{N}$. Unlike vision-based demonstrations, text-based ones require integrating action semantics and accumulating implicit state changes, as intermediate states are not directly observable.

Observation Sampling. Given a demonstration $\mathcal{D}$ and its execution video $\mathbf{V} = \{\mathbf{f}_k\}_{k=1}^{T}$, we construct observation–progress pairs $\mathcal{O} = \{(\mathbf{o}^i, \mathbf{p}^{*i})\}_{i=1}^{M}$. The video is partitioned into $N-1$ segments between consecutive key steps. For each segment, we sample intermediate frames at relative positions $\delta \in (0, 1)$ and assign progress via linear interpolation, $\mathbf{p}^* =$

$\mathbf{p}_j + \delta(\mathbf{p}_{j+1} - \mathbf{p}_j)$. We adopt two strategies: *interval sampling* for uniform coverage of intermediate progress and *boundary sampling* for finer resolution near step transitions. To evaluate viewpoint robustness, we further distinguish *same-view* and *cross-view* settings in the vision-based modality, where observations are captured from the same or different camera viewpoints as the demonstration.

Answerability Augmentation. To assess uncertainty awareness, we explicitly introduce *answerability*. *Answerable* samples correspond to coherent executions where progress can be inferred, while *unanswerable* samples arise when the observation is inconsistent with the demonstration and the correct output is `N/A`. Such cases are constructed by either modifying the demonstration or editing the observation while keeping the other fixed.

Data Source and Statistics. We build PROGRESS-BENCH on RoboMind (Wu et al., 2025b). As shown in Figure 3, the benchmark comprises approximately 240 task trajectories and 3325 sampled observations.

3 Towards Progress Reasoning

We frame progress reasoning as a human-inspired two-stage process. Given a demonstration and a partial observation, humans first perform *episodic retrieval* to identify a representative reference step as a coarse anchor, and then apply *mental simulation* to reason how the task state evolves from this anchor to the current observation. This formulation treats progress estimation as reasoning over a latent task trajectory, rather than matching observations to fixed timestamps.

3.1 Training-Free Approach

We instantiate this two-stage reasoning via structured prompting without parameter updates. The prompt enforces an explicit schema with four fields: `<ref_think>` (episodic retrieval reasoning), `<ref>` (retrieved reference step), `<score_think>` (mental simulation), and `<score>` (final progress estimate), which the model follows at inference time.

3.2 Training-Based Approach

We further adopt a training-based approach to explicitly teach episodic retrieval and mental simulation. We construct PROGRESSLM-45K from non-overlapping manipulation tasks with PROGRESS-BENCH, ensuring generalizable reasoning. Statistics are in Figure 3.

Cold-Start Supervised Fine-Tuning. We first perform supervised fine-tuning on PROGRESSLM-25K-COT to internalize the two-stage reasoning pattern. Each instance includes a demonstration $\mathcal{D}^i$, an observation $\mathbf{o}^i$, and a reasoning sequence $\mathbf{r}^{i*}$ containing ground-truth `<ref>` and `<score>`. Using guided reasoning completion, the model generates the intermediate fields `<ref_think>` and `<score_think>`. Training minimizes the autoregressive negative log-likelihood:

$$\mathcal{L}_{\text{SFT}} = -\frac{1}{N}\sum_{i=1}^{N} \log P_\theta(\mathbf{r}^{i*} \mid \mathcal{D}^i, \mathbf{o}^i). \quad (1)$$

Reinforcement Learning. To improve robustness and calibration, we apply a second-stage reinforcement learning procedure based on GRPO (Shao et al., 2024):

$$\mathcal{L}_{\text{RL}} = -\mathbb{E}_{\mathbf{r}\sim P_\theta(\mathbf{r}|\mathcal{D},\mathbf{o})}\big[R(\mathbf{r})\big], \quad (2)$$

where $R(\mathbf{r}) = \alpha R_{\text{format}} + \beta R_{\text{ref}} + \gamma R_{\text{score}}$ encourages structured reasoning, accurate reference retrieval, and precise progress estimation. Specifically, R_{ref} and R_{score} are defined as the negative relative errors between the predicted and ground-truth reference index and progress score, respectively. We train on 20K samples with $\alpha:\beta:\gamma = 1:6:3$.

4 Evaluation on PROGRESS-BENCH

Experimental Setup. We evaluate 14 VLMs (2B–72B), including GPT-5/mini (OpenAI, 2024), Qwen (Bai et al., 2025), and Intern models (Wang et al., 2025b), on PROGRESS-BENCH. Each model is tested with *direct prediction*, *training-free reasoning*, or *training-based reasoning* (PROGRESSLM-SFT/RL based on Qwen2.5-VL-3B). All training tasks are disjoint from the benchmark to ensure generalization.

Evaluation Design. We evaluate progress estimation using four metrics that capture pointwise accuracy, temporal consistency, and answerability awareness. **Normalized Score Error (NSE)** measures pointwise error on answerable samples. **Progress Rank Correla-**

Table 1: Results on *answerable* samples are reported using Normalized Score Error (NSE)↓, Progress Rank Correlation (PRC)↑, and Answerable False Rejection Rate (AFRR)↓. Best, Second, and Third results are highlighted; colored deltas indicate the effect of training-free reasoning (better or worse).

Model	Vision-based Demo			Text-based Demo			Micro Avg			Macro Avg		
	NSE↓	PRC↑	AFRR↓	NSE↓	PRC↑	AFRR↓	NSE↓	PRC↑	AFRR↓	NSE↓	PRC↑	AFRR↓
GPT-5	18.9 -0.6	89.4 -0.2	1.3 -0.2	23.6 -2.8	55.8 +4.3	7.0 +0.4	19.9 -1.0	82.5 +0.8	2.5 -0.1	21.3 -1.7	72.6 +2.0	4.2 +0.1
GPT-5-mini	20.7 +0.6	87.7 -0.6	0.4 -0.3	21.1 +1.3	55.2 +1.8	9.7 -5.1	20.8 +0.7	81.1 -0.1	2.3 -1.3	20.9 +1.0	71.4 +0.6	5.1 -2.7
Qwen2.5-VL-72B	27.0 -2.5	60.0 +18.2	5.9 -1.1	38.9 -12.5	41.6 +11.7	0.0 +29.4	29.4 -4.5	56.2 +16.7	4.7 +4.9	32.9 -7.5	50.8 +15.0	3.0 +14.2
Qwen2.5-VL-32B	38.9 -7.4	41.5 +30.2	0.0	42.6 -11.6	30.0 +20.8	0.0 +10.9	39.7 -8.3	39.2 +28.0	0.0	40.8 -9.5	35.8 +25.5	0.0 +5.5
Qwen2.5-VL-7B	34.0 +10.8	33.7 +6.3	28.3 +23.9	39.1 +2.9	20.5 +18.7	0.0 +20.1	35.0 +9.2	31.0 +9.6	22.5 +22.8	36.5 +6.8	27.1 +12.5	14.2 +22.0
Qwen2.5-VL-3B	32.2 +8.3	32.8 -1.9	0.02 +5.7	45.9 -2.7	7.5 +8.5	0.0 +17.1	35.0 +6.1	27.6 +0.6	0.02 +8.1	39.0 +2.8	20.2 +3.3	0.01 +11.4
Qwen3-VL-32B	20.2 +1.8	80.7 +6.1	0.3 +0.1	25.1 -1.3	51.9 +13.3	7.4 +0.0	21.2 +1.1	74.8 +7.6	1.7 +0.1	22.6 +0.2	66.3 +9.7	3.9 +0.0
Qwen3-VL-8B	23.5 +5.3	67.0 +8.1	0.0 +1.3	35.3 -10.2	34.2 +20.7	3.4 +15.3	25.9 +2.2	60.3 +11.0	0.7 +4.4	29.4 -2.4	50.6 +14.4	1.7 +8.4
Qwen3-VL-4B	30.6 +1.4	57.4 +10.3	0.0 +1.5	36.2 -7.0	38.1 +6.4	0.2 +9.5	31.8 -0.3	53.4 +9.5	0.04 +3.0	33.4 -2.8	47.8 +8.4	0.1 +5.6
Qwen3-VL-2B	64.5 -2.9	32.1 -7.6	5.2 +5.2	59.3 -8.6	NaN +24.1	12.5 +0.2	63.4 -4.1	NaN	6.7 +4.1	61.9 -5.7	NaN	8.9 +2.7
Intern3.5-VL-38B	35.2 +21.4	56.7 -31.0	37.1 -36.3	26.5 +4.2	23.3 +26.4	43.1 -28.5	33.4 +17.9	49.9 -22.4	38.3 -34.6	30.8 +12.8	40.0 -2.3	40.1 -32.4
Intern3.5-VL-14B	65.2 -4.7	-22.3 +42.8	0.2 +0.0	39.5 -5.0	10.3 +16.4	6.8 +0.0	60.0 -4.7	-15.6 +36.3	1.5 +0.0	52.3 -4.8	-6.0 +29.6	3.5 +0.0
Intern3.5-VL-4B	43.7 +12.0	18.0 -6.8	0.3 +0.0	37.0 -1.0	5.6 +10.9	10.9 +0.4	42.3 +9.3	15.5 -3.3	2.5 +0.1	40.4 +5.5	11.8 +2.1	5.6 +0.2
ProgressLM-3B-SFT	19.0	72.4	6.5	29.1	46.3	9.2	21.1	67.0	7.0	24.0	59.3	7.8
ProgressLM-3B-RL	13.8	90.1	8.5	21.2	63.9	5.6	15.3	84.8	7.9	17.5	77.0	7.0

tion (PRC) computes the Spearman rank correlation between predicted and ground-truth progress sequences along each trajectory, averaged across trajectories, with higher values close to 1 indicating better temporal ordering (Spearman, 1961). **Answerable False Rejection Rate (AFRR)** measures the fraction of answerable samples predicted as unanswerable, while **Unanswerable Detection Accuracy (UDA)** measures the fraction of unanswerable samples correctly identified.

4.1 Performance on Answerable Scenarios

We first evaluate model performance on *answerable* samples, where task progress is well-defined. Table 1 reports results under both vision-based and text-based demonstrations.

How well do current VLMs perform at progress estimation? ***Overall, current VLMs show limited and highly unstable progress estimation under direct prediction.*** Although strong models such as GPT-5 and Qwen2.5-VL-72B perform better than smaller counterparts, they remain highly sensitive to demonstration modality, with vision-based demonstrations consistently outperforming text-based ones. We further observe abnormally low, negative, or undefined PRC values for several models, indicating collapsed or distorted progress rankings rather than meaningful ordinal reasoning. As analyzed in Section 5.1, these failures stem from degenerate predicted score distributions, such as collapse to extreme or discrete values. Finally, some models (e.g., Intern3.5-VL-38B) exhibit extremely high AFRR even on answerable samples, reflecting overly conservative rejection behavior rather than calibrated uncertainty.

Does training-free progress reasoning help? ***Training-free reasoning yields conditional benefits and depends strongly on model capacity.*** Large models (e.g., GPT-5 and Qwen2.5-VL-72B/32B) gain moderate improvements, primarily in PRC and occasionally in NSE. In contrast, smaller models often see marginal or negative effects, including increased NSE or AFRR, suggesting that limited-capacity models may follow the reasoning format without genuinely improving progress understanding.

Does training-based progress reasoning help? ***Explicit training enables robust progress reasoning even at small scale.*** PROGRESSLM consistently improves over the base 3B model across all answerable metrics, with PROGRESSLM-RL-3B achieving the strongest macro-averaged NSE and PRC among all evaluated models. These gains demonstrate that effective progress reasoning is not solely driven by scale, but can be reliably learned through targeted supervision and optimization.

4.2 Robustness to Viewpoint Changes

To assess robustness to viewpoint changes, we decompose vision-based results into *same-view* and *cross-view* settings (Table 2).

Table 2: Same vs. cross-view performance on vision-based demonstrations. Best, Second, and Third results are highlighted; colored deltas indicate the effect of training-free reasoning (better or worse).

Model	Same-View			Cross-View			Delta (Same → Cross)		
	NSE↓	PRC↑	AFRR↓	NSE↓	PRC↑	AFRR↓	ΔNSE↓	ΔPRC↑	ΔAFRR↓
GPT-5	14.6 -0.4	89.4 +1.0	0.0	20.5 -0.6	89.4 -0.6	1.8 -0.3	+5.9 -0.2	0.0 -1.6	+1.8 -0.3
GPT-5-mini	19.8 +0.8	84.1 -0.6	0.4 -0.4	21.1 +0.6	89.1 -0.6	0.4 -0.2	+1.3 -0.2	+5.0 +0.0	0.0 +0.2
Qwen2.5-VL-72B	16.8 +5.0	83.9 -5.0	0.4 -0.5	30.9 -5.4	50.7 +26.4	13.4 -7.7	+14.1 -10.4	-33.2 +31.4	+13.0 -7.2
Qwen2.5-VL-32B	21.4 +6.2	72.9 +0.1	0.0	45.7 -12.7	29.4 +40.5	0.0	+24.3 -18.9	-43.5 +40.4	0.0 +0.0
Qwen2.5-VL-7B	27.3 +11.1	51.9 -7.6	30.6 +26.5	36.5 +10.6	26.7 +3.7	26.0 +21.4	+9.2 -0.5	-25.2 +11.3	-4.6 -5.1
Qwen2.5-VL-3B	29.2 +9.5	43.0 -11.1	9.9 +5.8	33.4 +7.9	28.9 -2.5	6.5 +4.4	+4.2 -1.6	-14.1 +8.6	-3.4 -1.4
Qwen3-VL-32B	15.9 +2.6	88.3 -8.0	0.0	21.9 +1.4	77.8 +11.0	0.4 -3.5	+6.0 -1.2	-10.5 +19.0	+0.4 -3.5
Qwen3-VL-8B	19.1 +5.6	81.7 -9.5	0.0 +1.3	25.2 +5.2	61.3 +16.0	0.0 +1.3	+6.1 -0.4	-20.4 +25.5	0.0 +0.0
Qwen3-VL-4B	21.6 +3.5	72.3 -3.2	0.0 +1.1	34.1 +0.4	51.6 +17.6	0.0 +1.6	+12.5 -3.1	-20.7 +20.8	0.0 +0.5
Qwen3-VL-2B	60.0 -1.6	35.0 -4.6	0.1 +9.8	66.2 -3.4	30.9 -12.6	0.0 +3.5	+6.2 -1.8	-4.1 -8.0	-0.1 -6.3
Intern3.5-VL-38B	27.6 +28.4	74.8 -51.5	0.5 +0.1	38.1 +18.6	49.7 -17.9	51.3 -50.4	+10.5 -9.8	-25.1 +33.6	+50.8 -50.5
Intern3.5-VL-14B	62.1 +0.0	24.2 -4.5	0.0 +0.0	66.4 -6.4	-40.3 +65.4	0.0	+4.3 -6.4	-64.5 +69.9	0.0 +0.0
Intern3.5-VL-4B	44.4 +9.6	33.1 -18.9	0.7 -0.4	43.5 +12.8	12.1 -8.6	0.2 -0.2	-0.9 +3.2	-21.0 +10.3	-0.5 +0.2
ProgressLM-3B-SFT	15.5	84.0	0.6	20.4	67.8	8.8	+4.9	-16.2	+8.2
ProgressLM-3B-RL	10.3	93.5	0.1	15.2	88.8	11.7	+4.9	-4.7	+11.6

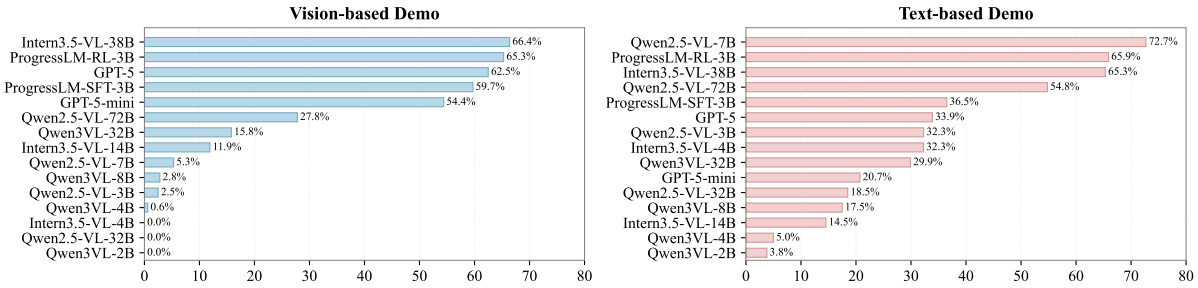


Figure 4: Unanswerable Detection Accuracy (UDA) across models under two settings.

How do current VLMs handle viewpoint changes? ***Most VLMs degrade substantially under cross-view observations.*** Across models, cross-view settings yield higher NSE and lower PRC than same-view ones, with especially large drops for small and medium-sized models. This pattern suggests that many VLMs rely heavily on viewpoint-dependent visual similarity rather than viewpoint-invariant progress reasoning.

Does progress reasoning improve cross-view robustness? ***Training-free progress reasoning provides limited and conditional gains.*** Its effectiveness strongly depends on baseline capability: weaker models benefit little, while strong models (e.g., GPT-5) show only modest cross-view improvements, often at the expense of same-view performance. When gains occur, they primarily appear in PRC rather than NSE, indicating improved temporal ordering rather than pointwise accuracy. ***In contrast, robust cross-view generalization emerges only through explicit training.*** PROGRESSLM-3B-RL consistently exhibits smaller same-view to cross-view gaps, demonstrating improved robustness beyond surface-level visual correspondence.

4.3 Unanswerable Case Recognition

Can models recognize when progress estimation is not possible? ***Most VLMs fail to reliably recognize unanswerable cases.*** As shown in Figure 4, most models still produce progress scores even when the input is inherently ambiguous, indicating limited awareness of unanswerability. ***In contrast,*** **PROGRESSLM** ***consistently identifies unanswerable cases under both vision-based and text-based demonstrations, avoiding forced predictions.*** This behavior is further strengthened by reinforcement learning, with PROGRESSLM-3B-RL achieving the highest or near-highest unanswerable recognition accuracy. However, strong unan-

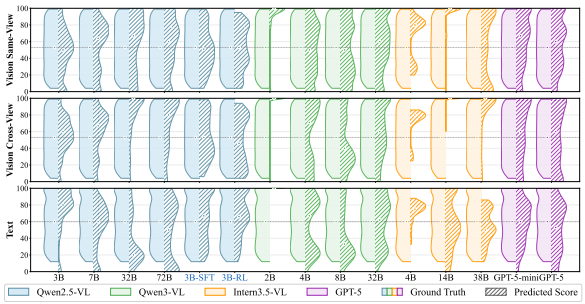


Figure 5: Distribution of predicted progress scores.

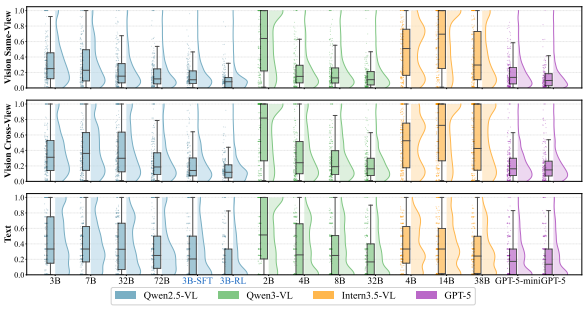


Figure 6: Raincloud plots of per-sample normalized score prediction error across models.

swerable detection alone is insufficient. For example, INTERNVL-3.5-38B attains relatively high UDA but exhibits an extremely high Answerable False Rejection Rate (AFRR) (Table 1), indicating overly conservative behavior.

5 Further Analysis

5.1 Predicted Score Distribution

What patterns emerge in predicted progress score distributions? Predicted scores often cluster around specific values rather than varying smoothly. We observe four recurring patterns in Figure 5: ***(i) single-peak collapse*** at extreme values (e.g., 0% or 100%); ***(ii) multi-peak clustering*** around a few heuristic anchors; ***(iii) central-peaked distributions*** concentrated near $\sim 50\%$, reflecting default or uncertain responses; and ***(iv) smooth continuous distributions*** spanning $[0, 100\%]$, indicating sensitivity to intermediate task states. Notably, PROGRESSLM-3B-SFT and PROGRESSLM-3B-RL consistently exhibit the fourth pattern, whereas most base VLMs fall into the first three. These distributional failures directly explain unstable or distorted rank correlations, underscoring that fine-grained progress reasoning emerges only through explicit learning.

5.2 Per-sample Error Distribution

What do per-sample error distributions reveal? ***Smaller models exhibit unstable errors, while explicit progress learning substantially improves robustness.*** In Figure 6, smaller models show broad, heavy-tailed error distributions, reflecting frequent large deviations and unstable progress estimates, consistent with their collapsed or spiky score distributions. Larger models produce more concentrated errors closer to zero. Notably, both PROGRESSLM-3B-SFT and PROGRESSLM-3B-RL markedly tighten the error distribution relative to the base 3B model, with PROGRESSLM-3B-RL especially suppressing extreme-error cases. These results indicate that explicitly learned progress reasoning improves not only average accuracy but also per-sample robustness.

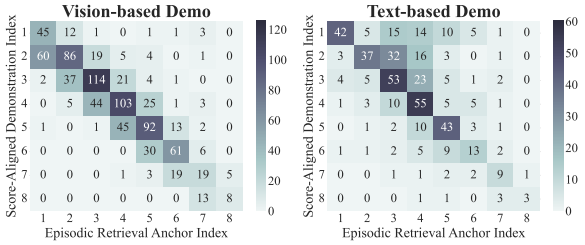


Figure 7: Coupling between the two stages progress reasoning of PROGRESSLM.

5.3 Coupled Progress Reasoning

Are the two reasoning stages truly coupled? ***Yes—the first-stage anchor directly constrains second-stage progress estimation.*** Figure 7 analyzes the relationship between the *Episodic Retrieval Anchor Index* (the step retrieved in episodic retrieval) and the *Score-Aligned Demonstration Index* (the step whose annotated progress best matches the predicted score). If the two stages are coupled, these indices should align, yielding a near-diagonal pattern. The strong diagonal concentration observed for our training-based model confirms that anchor retrieval is not auxiliary, but actively guides fine-grained progress estimation via second-stage *mental simulation*.

5.4 Implicit State Accumulation

Why are text-based demonstrations harder than vision-based ones? ***Text-based demonstrations require implicit state accumula-***

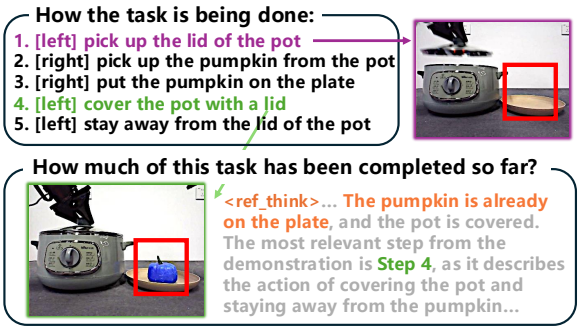


Figure 8: Illustration of implicit state accumulation required by text-based demonstrations.

tion rather than direct state matching. Unlike vision-based demonstrations that explicitly show complete world states, text-based demonstrations describe actions whose effects must be integrated over time. As illustrated in Figure 8, Step 1 and Step 4 both involve interacting with the pot lid, yet differ in an implicit state variable: whether the pumpkin has already been removed and placed on the plate. Disambiguating these steps requires integrating the intervening actions (Steps 2 and 3), highlighting that text-based progress estimation depends on tracking latent state evolution rather than surface-level action semantics.

6 Related Work

Progress Estimation. Early progress estimation methods predominantly rely on task-specific or expert models trained within fixed tasks or environments (Yang et al., 2024; Chen et al., 2025; Ma et al., 2024a), which limits their ability to generalize beyond the training distribution. Some approaches estimate progress by measuring distances in latent feature space (Ma et al., 2022, 2023; Escontrela et al., 2023; Lee et al., 2021), but they struggle to model fine-grained intermediate progress. More recently, VLMs have been applied to progress estimation, but typically through indirect formulations (Alakuijala et al., 2025; Ma et al., 2024b; Zhai et al., 2025). For example, progress is inferred via trajectory reordering by shuffling frames (Ma et al., 2024b), or by aggregating pairwise relative progress comparisons (Zhai et al., 2025). In these settings, progress estimates are not independent, but strongly coupled to other predictions within the same trajectory, causing each estimate to depend on the entire sequence context. In contrast, humans can infer task progress from a single observation by reasoning over underlying task dynamics, highlighting progress estimation as a long-horizon and dynamic reasoning problem.

Progress Reasoning in VLMs. Recent advances in VLMs have substantially improved static visual reasoning capabilities from single or multiple images (Zhang et al., 2025; Lee et al., 2025; Yuan et al., 2025a,b; Pi et al., 2024; Xing et al., 2025; Cai et al., 2025), largely focusing on snapshot-level perception. However, dynamic reasoning under partial observation requires models to reason over long-horizon task evolution, infer latent state transitions, and maintain an implicit world model (Yin et al., 2025; Wang et al., 2025a). Progress reasoning naturally embodies these challenges, as it demands fine-grained understanding of intermediate task states from a single observation. Nevertheless, most existing VLM-based approaches focus on *coarse task completion judgments*, often reducing progress estimation to binary decisions and overlooking whether models can reason about *intermediate progress from a single observation* (Fan et al., 2022; Cui et al., 2022; Wang et al., 2024; Lu et al., 2025; Duan et al., 2024; Luo et al., 2025; Dai et al., 2025). This gap motivates our work, which studies progress estimation as a structured reasoning capability in VLMs rather than a heuristic or static prediction.

7 Conclusion

We study progress estimation as a long-horizon, dynamic reasoning problem beyond static visual understanding. We introduce PROGRESS-BENCH to systematically evaluate progress reasoning from a single observation under controlled variations of modality, viewpoint, and answerability. Experiments on 14 VLMs show that existing models struggle with this task, exhibiting strong sensitivity to modality and viewpoint changes, degenerate progress predictions, and weak handling of unanswerable cases. Our analyses expose systematic failure modes in existing VLMs and show that robust progress estimation emerges only when coarse anchor retrieval and fine-grained reasoning are explicitly learned.

8 Limitations

While this work provides a systematic study of progress reasoning in VLMs, it has several limitations. PROGRESS-BENCH focuses on robotic manipulation tasks, where progress typically follows clear and monotonic structures. This controlled setting is well suited for isolating progress reasoning behavior, but the conclusions may not directly generalize to more open-ended scenarios involving ambiguous goals, non-monotonic progress, or subjective notions of completion. In addition, our training-based model, PROGRESSLM-3B, is trained on a curated set of manipulation tasks that, while disjoint from the benchmark, still share similar structural properties. Although we observe strong generalization across unseen tasks within this domain, extending progress reasoning to substantially different task families may require further data curation or adaptation strategies. We leave these directions to future work and hope our study serves as a foundation for broader investigations into progress reasoning with VLMs.

References

Minttu Alakuijala, Reginald McLean, Isaac Woungang, Nariman Farsad, Samuel Kaski, Pekka Marttinen, and Kai Yuan. 2025. Video-language critic: Transferable reward functions for language-conditioned robotics. *Transactions on Machine Learning Research (TMLR)*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Zhipeng Cai, Ching-Feng Yeh, Hu Xu, Zhuang Liu, Gregory Meyer, Xinjie Lei, Changsheng Zhao, Shang-Wen Li, Vikas Chandra, and Yangyang Shi. 2025. Depthlm: Metric depth from vision language models. *arXiv preprint arXiv:2509.25413*.

Qianzhong Chen, Justin Yu, Mac Schwager, Pieter Abbeel, Fred Shentu, and Philipp Wu. 2025. Sarm: Stage-aware reward modeling for long horizon robot manipulation. *Preprint*, arXiv:2509.25358.

Yuchen Cui, Scott Niekum, Abhinav Gupta, Vikash Kumar, and Aravind Rajeswaran. 2022. Can foundation models perform zero-shot task specification for robot manipulation? In *Learning for dynamics and control conference*, pages 893–905. PMLR.

Yinpei Dai, Jayjun Lee, Nima Fazeli, and Joyce Chai. 2025. Racer: Rich language-guided failure recovery policies for imitation learning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15657–15664. IEEE.

Tri Dao. 2023. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*.

Jiafei Duan, Wilbert Pumacay, Nishanth Kumar, Yi Ru Wang, Shulin Tian, Wentao Yuan, Ranjay Krishna, Dieter Fox, Ajay Mandlekar, and Yijie Guo. 2024. Aha: A vision-language-model for detecting and reasoning over failures in robotic manipulation. *arXiv preprint arXiv:2410.00371*.

Alejandro Escontrela, Ademi Adeniji, Wilson Yan, Ajay Jain, Xue Bin Peng, Ken Goldberg, Youngwoon Lee, Danijar Hafner, and Pieter Abbeel. 2023. Video prediction models as rewards for reinforcement learning. *Advances in Neural Information Processing Systems*, 36:68760–68783.

Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. 2022. Minedojo: Building open-ended embodied agents with internet-scale knowledge. *Advances in Neural Information Processing Systems*, 35:18343–18362.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Young-Jun Lee, Byung-Kwan Lee, Jianshu Zhang, Yechan Hwang, Byungsoo Ko, Han-Gyu Kim, Dongyu Yao, Xuankun Rong, Eojin Joo, Seung-Ho Han, and 1 others. 2025. Multiverse: A multi-turn conversation benchmark for evaluating large vision and language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 708–719.

Youngwoon Lee, Andrew Szot, Shao-Hua Sun, and Joseph J Lim. 2021. Generalizable imitation learning from observation via inferring goal proximity. *Advances in neural information processing systems*, 34:16118–16130.

Weifeng Lu, Minghao Ye, Zewei Ye, Ruihan Tao, Shuo Yang, and Bo Zhao. 2025. Robofac: A comprehensive framework for robotic failure analysis and correction. *arXiv preprint arXiv:2505.12224*.

Zhen Luo, Yixuan Yang, Yanfu Zhang, and Feng Zheng. 2025. Roboreflect: A robotic reflective reasoning framework for grasping ambiguous-condition objects. *arXiv preprint arXiv:2501.09307*.

Jason Ma, William Liang, Hung-Ju Wang, Yuke Zhu, Linxi Fan, Osbert Bastani, and Dinesh Jayaraman. 2024a. Dreureka: Language model guided sim-to-real transfer. RSS.

Yecheng Jason Ma, Joey Hejna, Chuyuan Fu, Dhruv Shah, Jacky Liang, Zhuo Xu, Sean Kirmani, Peng Xu, Danny Driess, Ted Xiao, and 1 others. 2024b. Vision language models are in-context value learners. In *The Thirteenth International Conference on Learning Representations*.

Yecheng Jason Ma, Vikash Kumar, Amy Zhang, Osbert Bastani, and Dinesh Jayaraman. 2023. Liv: Language-image representations and rewards for robotic control. In *International Conference on Machine Learning*, pages 23301–23320. PMLR.

Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. 2022. Vip: Towards universal visual reward and representation via value-implicit pre-training. *arXiv preprint arXiv:2210.00030*.

Philipp Moritz, Robert Nishihara, Stephanie Wang, Alexey Tumanov, Richard Liaw, Eric Liang, Melih Elibol, Zongheng Yang, William Paul, Michael I Jordan, and 1 others. 2018. Ray: A distributed framework for emerging {AI} applications. In *13th USENIX symposium on operating systems design and implementation (OSDI 18)*, pages 561–577.

OpenAI. 2024. Gpt-5 system card. https://openai.com/index/gpt-5-system-card/.

Renjie Pi, Jianshu Zhang, Jipeng Zhang, Rui Pan, Zhekai Chen, and Tong Zhang. 2024. Image textualization: An automatic framework for creating accurate and detailed image descriptions. *arXiv preprint arXiv:2406.07502*.

Daniel L Schacter, Donna Rose Addis, and Randy L Buckner. 2008. Episodic simulation of future events: Concepts, data, and applications. *Annals of the New York Academy of Sciences*, 1124(1):39–60.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Charles Spearman. 1961. The proof and measurement of association between two things.

Leitian Tao, Ilia Kulikov, Swarnadeep Saha, Tianlu Wang, Jing Xu, Yixuan Li, Jason E Weston, and Ping Yu. 2025. Hybrid reinforcement: When reward is sparse, it's better to be dense. *arXiv preprint arXiv:2510.07242*.

Kangrui Wang, Pingyue Zhang, Zihan Wang, Yaning Gao, Linjie Li, Qineng Wang, Hanyang Chen, Chi Wan, Yiping Lu, Zhengyuan Yang, and 1 others. 2025a. Vagen: Reinforcing world model reasoning for multi-turn vlm agents. *arXiv preprint arXiv:2510.16907*.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, and 1 others. 2025b. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*.

Yufei Wang, Zhanyi Sun, Jesse Zhang, Zhou Xian, Erdem Biyik, David Held, and Zackory Erickson. 2024. Rl-vlm-f: Reinforcement learning from vision language foundation model feedback. *arXiv preprint arXiv:2402.03681*.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, and 20 others. 2025a. Qwen-image technical report. *Preprint*, arXiv:2508.02324.

Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, Zhuqin Yang, Meng Li, Yinuo Zhao, Zhiyuan Xu, Guang Yang, and 1 others. 2025b. Robomind: Benchmark on multi-embodiment intelligence normative data for robot manipulation. In *Robotics: Science and Systems (RSS) 2025*. Robotics: Science and Systems Foundation.

Shuo Xing, Peiran Li, Yuping Wang, Ruizheng Bai, Yueqi Wang, Chan-Wei Hu, Chengxuan Qian, Huaxiu Yao, and Zhengzhong Tu. 2025. Re-align: Aligning vision language models via retrieval-augmented direct preference optimization. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 2379–2397.

Daniel Yang, Davin Tjia, Jacob Berg, Dima Damen, Pulkit Agrawal, and Abhishek Gupta. 2024.

Rank2reward: Learning shaped reward functions from passive video. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2806–2813. IEEE.

Baiqiao Yin, Qineng Wang, Pingyue Zhang, Jianshu Zhang, Kangrui Wang, Zihan Wang, Jieyu Zhang, Keshigeyan Chandrasegaran, Han Liu, Ranjay Krishna, and 1 others. 2025. Spatial mental modeling from limited views. In *Structural Priors for Vision Workshop at ICCV'25*.

Zhenlong Yuan, Chengxuan Qian, Jing Tang, Rui Chen, Zijian Song, Lei Sun, Xiangxiang Chu, Yujun Cai, Dapeng Zhang, and Shuo Li. 2025a. Autodrive-r2: Incentivizing reasoning and self-reflection capacity for vla model in autonomous driving. *arXiv preprint arXiv:2509.01944*.

Zhenlong Yuan, Xiangyan Qu, Chengxuan Qian, Rui Chen, Jing Tang, Lei Sun, Xiangxiang Chu, Dapeng Zhang, Yiwei Wang, Yujun Cai, and 1 others. 2025b. Video-star: Reinforcing open-vocabulary action recognition with tools. *arXiv preprint arXiv:2510.08480*.

Shaopeng Zhai, Qi Zhang, Tianyi Zhang, Fuxian Huang, Haoran Zhang, Ming Zhou, Shengzhe Zhang, Litao Liu, Sixu Lin, and Jiangmiao Pang. 2025. A vision-language-action-critic model for robotic real-world reinforcement learning. *arXiv preprint arXiv:2509.15937*.

Jianshu Zhang, Dongyu Yao, Renjie Pi, Paul Pu Liang, and Yi R Fung. 2025. Vlm2-bench: A closer look at how well vlms implicitly link explicit matching visual cues. *arXiv preprint arXiv:2502.12084*.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, and 1 others. 2023. Pytorch fsdp: experiences on scaling fully sharded data parallel. *arXiv preprint arXiv:2304.11277*.

Yaowei Zheng, Junting Lu, Shenzhi Wang, Zhangchi Feng, Dongdong Kuang, and Yuwen Xiong. 2025. Easyr1: An efficient, scalable, multi-modality rl training framework.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. *arXiv preprint arXiv:2403.13372*.

Appendix

A Data Construction Details **12**
- A.1 Text Unanswerable Data 12
- A.2 Vision-Based Unanswerable Data 12
- A.3 Chain-of-Thought Generation . 13
- A.4 Human Bench Data 14

B Experimental Settings **15**
- B.1 Model Inference. 15
- B.2 Text-Based Unanswerable Data Generation. 15
- B.3 Vision-Based Unanswerable Data Generation. 15
- B.4 Chain-of-Thought Data Generation 15
- B.5 Supervised Fine-Tuning. 16
- B.6 Reinforcement Learning. 16

C Supplementary Results and Analysis **16**
- C.1 Vision-Based Demo Case Studies. 16
- C.2 Text-Based Demo Case Studies. 17
- C.3 Analysis of Coupled Progress Two-Stage Reasoning 19
- C.4 In the Wild Generalization Analysis 20
- C.5 Unanswerable Case Recognition 22

D Prompts **23**
- D.1 Vision-based Demo 23
- D.2 Text-based Demo 23
- D.3 Vision-based Chain-of-Thought Prompt 23
- D.4 Text-based Chain-of-Thought Prompt 23
- D.5 Unanswerable Vision-based Sample Generation 23
- D.6 Unanswerable Text-Based Sample Generation 23

A Data Construction Details

A.1 Text Unanswerable Data

We construct text negative samples by replacing key objects in task instructions to create misalignment between the visual observation and textual description. Given the task goal G, step-by-step instructions $\{s_1, s_2, \ldots, s_n\}$, and optionally a reference image I, we employ Qwen2.5-VL-72B (Bai et al., 2025) to generate modified instructions where the main manipulated object is replaced with a different object. The model is instructed to:

1. Analyze the input state-to-estimate image and identify objects that could serve as plausible replacements
2. Replace the target object in both the task goal and all step-by-step instructions
3. Preserve the original sentence structure, action verbs, and spatial markers (*e.g.*, `[left]`, `[right]`, `[towards]`)

The model outputs the modified task goal and instructions in a structured XML format, as shown in Table 9. This approach ensures that the edited instructions remain grammatically coherent and physically plausible while creating a clear mismatch between the visual scene and the textual description.

A.2 Vision-Based Unanswerable Data

We construct visual unanswerable samples through a three-stage pipeline to evaluate model robustness against adversarial visual perturbations.

Stage 1: Edit Prompt Generation. Given an input image I along with the task goal G and step-by-step instructions $\{s_1, s_2, \ldots, s_n\}$, we first identify the corresponding instruction s_k that matches the current image state. We then employ Qwen2.5-VL-72B (Bai et al., 2025) to generate an editing prompt that would cause the instruction to be violated. The model is provided with a structured prompt containing: (1) the task goal and complete step-by-step instructions, (2) the input image to be edited, and (3) the specific instruction s_k corresponding to the current image. The model is instructed to select one editing strategy from three predefined options:

1. **Color Change**: Altering the color of critical objects (*e.g.*, changing a red apple to green)
2. **Object Replacement**: Replacing the target object with a semantically different object (*e.g.*, replacing an egg with an orange)
3. **Occlusion/Removal**: Hiding or removing key objects from the scene

The model first reasons about which strategy would most effectively violate the instruction while maintaining visual realism, then

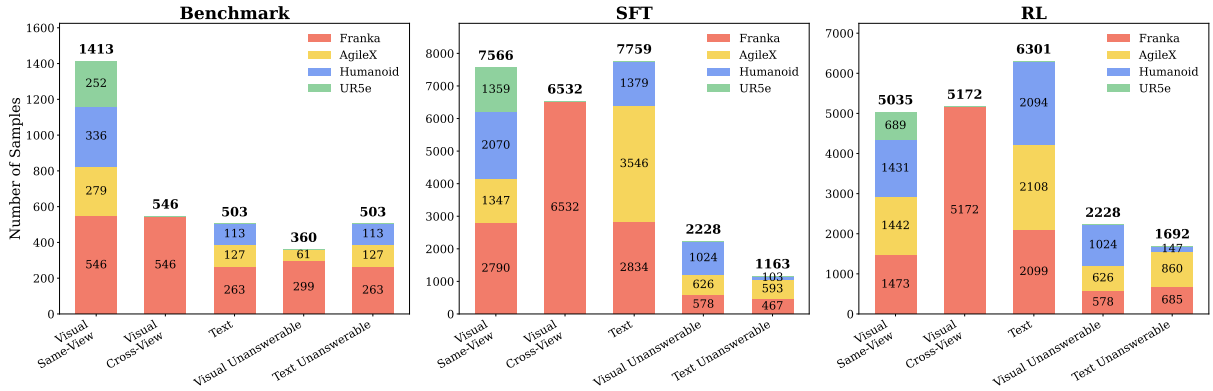


Figure 9: **Data distribution statistics across Benchmark, SFT, and RL splits.** This figure shows the distribution of samples produced by our data construction pipeline across Benchmark, Supervised Fine-Tuning (SFT), and Reinforcement Learning (RL) stages. Samples are organized by demonstration–observation setting (Visual Same-View, Visual Cross-View, Text, Visual Unanswerable, Text Unanswerable), with stacked bars denoting different robot platforms. Our constructed dataset spans four heterogeneous robotic platforms, including single-arm robot (Franka Emika Panda, UR5e), dual-arm robot (AgileX Cobot Magic V2.0), and humanoid robot (X-Humanoid Tien Kung), enabling evaluation and training across diverse embodiments.

outputs a concise editing prompt (maximum 20 words) in a structured XML format containing the reasoning process, selected strategy, and the final editing instruction. The complete prompt for is provided in Table 8.

Stage 2: Image Editing. We apply the generated editing prompts to the original images using Qwen-Image-Edit (20B) (Wu et al., 2025a), a diffusion-based image editing model built upon the Diffusers library. The model takes the original image I and the editing prompt p as input, and generates an edited image I' through an iterative denoising process.

Stage 3: Human Filtering. We develop a web-based annotation platform using **Gradio Platform** (As shown in Figure 20) to perform human quality control on the edited images. Annotators are presented with the edited image alongside the task goal, step-by-step instructions, editing strategy, and editing prompt. They assess whether the edit successfully violates the corresponding instruction while maintaining visual realism and naturalness. Each sample is labeled as either "YES" (keep) or "NO" (discard). Through this rigorous filtering process, we retain 23.5% of the edited images that meet our quality criteria, resulting in a high-quality visual negative dataset.

A.3 Chain-of-Thought Generation

To construct high-quality Chain-of-Thought (CoT) training data for progress estimation, we employ a *ground-truth guided generation* approach using Qwen2.5-VL-72B. Unlike conventional CoT distillation where models generate reasoning freely, our method constrains the generation by providing partial ground-truth answers—specifically the reference step index and final progress score—while requiring the model to synthesize coherent reasoning chains that justify these conclusions.

The generation process operates on two demonstration modalities: (1) **Text-Based Demo CoT**: Given a task goal and step-by-step textual instructions (e.g., "Step 1. reach for the power bank; Step 2. insert the battery..."), the model receives the current state image along with the ground-truth reference step (which text step most closely matches the current state) and progress score. The model generates reasoning in two phases: `<ref_think>` explains why the given reference step is most relevant to the current image, and `<score_think>` justifies how comparing the current state with the reference step yields the given progress score. (2) **Visual-Based Demo CoT**: Given a sequence of demonstration frames with associated progress values (e.g., "<image> 0% <image> 25% <image> 50% <image> 75% <im-

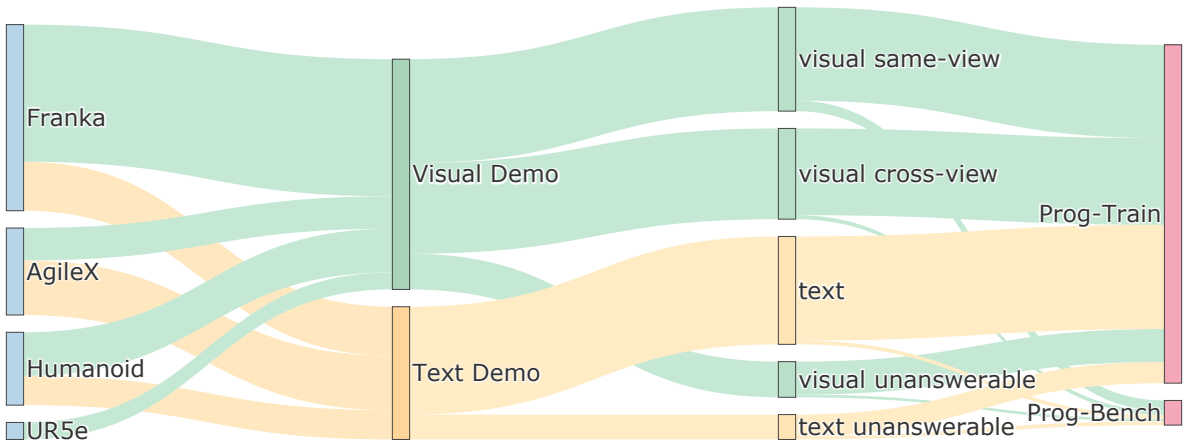


Figure 10: **Overview of the data construction pipeline.** This Sankey diagram illustrates how raw manipulation trajectories from four heterogeneous robotic platforms (Franka, AgileX, Humanoid, and UR5e) are transformed through our data construction process. Demonstrations are first organized into visual and text modalities, then further expanded into multiple demonstration–observation settings, including visual same-view, visual cross-view, text, as well as visual and text unanswerable cases. The resulting data are finally allocated to **PROGRESSLM-45K** for model training and **PROGRESSLM-BENCH** for evaluation, highlighting the unified yet diversified pipeline that supports generalizable progress reasoning across embodiments, modalities, and answerability conditions.

age> 100%"), the model receives the current state image and ground-truth annotations, then generates analogous reasoning explaining the visual comparison between the current state and demonstration frames.

This constrained generation strategy ensures that the synthesized CoT data exhibits consistent reasoning patterns aligned with correct answers, avoiding the noise introduced by freely-generated reasoning that may lead to incorrect conclusions.

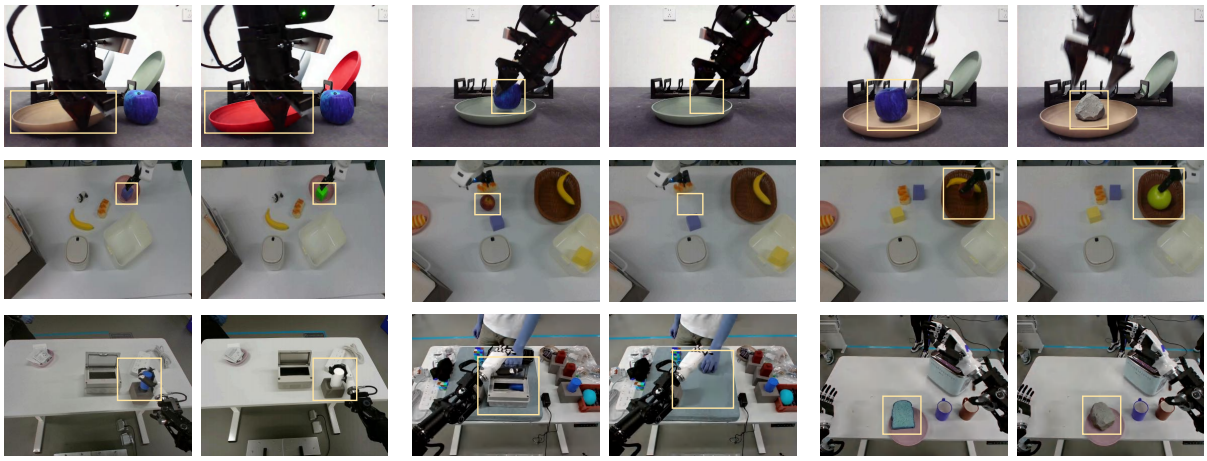
A.4 Human Bench Data

To evaluate model generalization capabilities in real-world scenarios, we construct a human activity benchmark through manual data collection.

Dataset Construction. Unlike existing robotic manipulation datasets collected from teleoperation systems, our benchmark captures *human hand manipulation* activities in uncontrolled environments. Human annotators perform various manipulation tasks while recording video sequences from a top-down camera view. For each task, we collect: (1) a visual demonstration sequence showing the complete task execution, and (2) multiple test frames sampled from different execution instances of the same task. This setup creates natural domain shift between the demonstration and test frames, as different human performers exhibit variations in hand appearance, manipulation style, and environmental conditions.

Task Categories. The human activity benchmark comprises 587 samples spanning 129 unique task goals across six manipulation categories: (1) **Pushing**: moving objects toward target objects or positions (*e.g.*, "pushing a red cup to a rubic cube"); (2) **Pick-and-Place**: placing objects into containers such as baskets (*e.g.*, "putting a jar into the blue basket"); (3) **Placing on Surface**: positioning objects on flat surfaces like plates (*e.g.*, "placing an orange on a plate"); (4) **Container Insertion**: inserting objects into enclosed containers (*e.g.*, "place the Rubik's cube inside the metal container"); (5) **Stacking**: placing objects on top of other objects (*e.g.*, "putting a rubik's cube on the top of the box"); (6) **Peg Manipulation**: Tower of Hanoi style block-on-peg tasks (*e.g.*, "place the blue block onto the middle red peg").

In-the-Wild Challenges. This benchmark introduces several challenges that push model capabilities beyond controlled laboratory settings: (1) **Domain Gap**: models trained on robotic manipulation must generalize to human hand appearances and motion patterns; (2) **Environmental Variation**: uncontrolled



(a) Color Change (b) Object Removal (c) Object Replacement

Figure 11: **Case Visualization of visual unanswerable samples construction via Image Editing**. To test whether models can recognize ill-defined progress, we construct visual unanswerable samples by breaking the semantic consistency between demonstrations and observations while preserving realism. Given an image at a specific manipulation step, we edit the key object using three strategies: (a) Color Change, altering object appearance; (b) Object Removal, eliminating the critical object; and (c) Object Replacement, substituting it with an incompatible one. As shown across diverse manipulation scenarios, these edits invalidate progress estimation and require models to correctly output N/A rather than relying on superficial visual matching.

lighting, backgrounds, and object arrangements; (3) **Execution Diversity**: different human performers exhibit distinct manipulation styles and trajectories; (4) **Viewpoint Consistency**: top-down camera views differ from typical robotic camera setups.

B Experimental Settings

B.1 Model Inference.

For fair comparison, we adopt unified inference settings across all evaluated open-source vision-language models, including Qwen3-VL (4B, 8B, 32B), Qwen2.5-VL (3B, 7B, 32B, 72B), and InternVL3.5 (4B, 14B, 38B). Specifically, we set temperature to 0.6, top-p to 0.9, and top-k to 50 for all models. The maximum number of generated tokens is set to 4,096 for inference. All models use bfloat16 precision and Flash Attention 2 for efficient inference.

B.2 Text-Based Unanswerable Data Generation.

We use Qwen2.5-VL-72B (Bai et al., 2025) for text object replacement generation. The model is distributed across 4 NVIDIA H100 (80GB) GPUs using model parallelism. We set temperature to 0.7, top-p to 0.9, top-k to 50, and maximum output tokens to 30,000.

B.3 Vision-Based Unanswerable Data Generation.

Edit Prompt Generation. We use Qwen2.5-VL-72B (Bai et al., 2025) for generating editing prompts. The model is loaded in bfloat16 precision with Flash Attention 2 (Dao, 2023) enabled. We set temperature to 0.7 to encourage diverse editing strategies, top-p to 0.9, top-k to 50, and maximum output tokens to 1,024. The image resolution is constrained between 1,003,520 and 4,014,080 pixels. Inference is conducted on 4 NVIDIA H100 (80GB) GPUs with a batch size of 8 per GPU.

Image Editing. We use Qwen-Image-Edit (20B) (Wu et al., 2025a) with 4 NVIDIA H100 (80GB) GPUs for image editing. Each GPU processes one image at a time (batch size of 1) due to the memory-intensive nature of diffusion models. The number of diffusion inference steps is set to 50, and the classifier-free guidance scale is set to 4.0. We use a single space character as the negative prompt and a fixed random seed of 42 for reproducibility. All editing is performed in bfloat16 mixed precision.

B.4 Chain-of-Thought Data Generation

We use Qwen2.5-VL-72B (Bai et al., 2025) for CoT data generation with temperature 0.6, top-p 0.9, top-k 50, and maximum new tokens

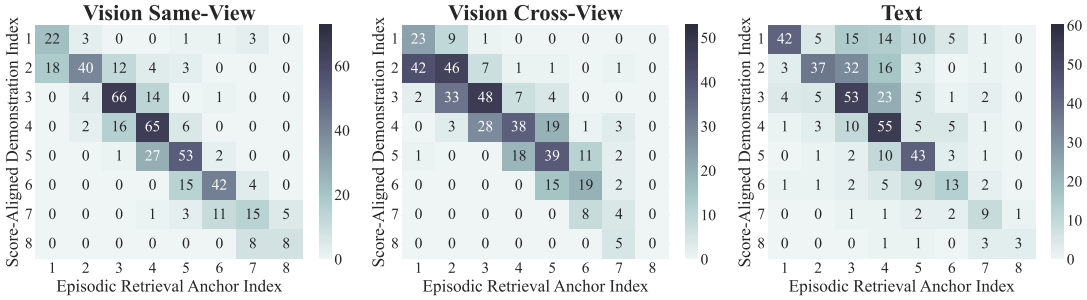


Figure 12: **Diagnostic analysis of coupled progress reasoning.** Heatmaps show the relationship between the episodic retrieval anchor index (x-axis) and the score-aligned demonstration index (y-axis) under Vision Same-View, Vision Cross-View, and Text settings. A strong diagonal indicates tight coupling between episodic retrieval and progress estimation. While coupling is strongest in the same-view setting and gradually weakens under cross-view and text conditions, the persistent diagonal structure across all settings demonstrates that progress estimation is consistently anchored to episodic retrieval rather than performed as direct regression.

set to 4096 to accommodate extended reasoning chains. The model is distributed across 4 GPUs using model parallelism. For text-demo generation, we use batch size 40; for visual-demo generation with multiple input images per sample, we use batch size 8 to accommodate the increased memory requirements.

B.5 Supervised Fine-Tuning.

We conduct Supervised Fine-Tuning (SFT) using the LLaMA-Factory framework (Zheng et al., 2024). We adopt LoRA (Hu et al., 2022) with rank 8 applied to all linear layers for parameter-efficient fine-tuning. The learning rate is set to 1×10^{-4} with a cosine scheduler and 10% warmup ratio. We use a per-device batch size of 2 with gradient accumulation steps of 8, resulting in an effective batch size of 64 on 4 NVIDIA H100 (80GB) GPUs. Models are trained for 2 epochs with BFloat16 mixed precision. The maximum sequence length is set to 30,000 tokens to accommodate multi-modal inputs with multiple images.

B.6 Reinforcement Learning.

We perform RL training using EasyR1 (Zheng et al., 2025), a clean version of verl (Tao et al., 2025) with Group Relative Policy Optimization (GRPO) (Shao et al., 2024) as the advantage estimator. The actor learning rate is 1×10^{-6} with AdamW optimizer. We set the KL divergence coefficient to 0.01 using the low-variance KL penalty. The global batch size is set to 64, with micro batch size of 1 per device for both policy update and experience collection. For each prompt, we generate $n = 16$ rollout samples with temperature 0.6 and top-p of 0.9. The maximum prompt length is 20,000 tokens and maximum response length is 8,192 tokens. We use Fully Sharded Data Parallel (FSDP) (Zhao et al., 2023) and Ray (Moritz et al., 2018) for distributed training across 4 nodes with 4 NVIDIA H100 (80GB) GPUs per node (16 GPUs in total). The model is trained for 2 epochs with totally 20428 samples, requiring approximately 23 hours in total. We also using vLLM (Kwon et al., 2023) serving as the generation engine with 60% GPU memory utilization.

C Supplementary Results and Analysis

C.1 Vision-Based Demo Case Studies.

We provide qualitative case studies to further illustrate how vision-based demonstrations support coupled progress reasoning under both same-view and cross-view settings. These examples complement the quantitative results reported in the main paper and offer mechanistic insights into how episodic retrieval and progress estimation interact in practice.

Same-view reasoning with fine-grained state alignment. In the same-view case, the demonstration frames and the state to estimate share a consistent viewpoint, enabling direct visual alignment. As shown in the plate-stacking example, the model retrieves a demonstration step corresponding to a near-completion stage, where the plates are almost

Table 3: Human Bench: Comparison of in-the-wild model generalization performance on Visual and Textual Input evaluations. Best Second Best Third Best . NSE↓, PRC↑, AFRR↓. (think better or worse).

Model	Vision-based Demo			Text-based Demo			Average		
	NSE	PRC	AFRR	NSE	PRC	AFRR	NSE	PRC	AFRR
Qwen2.5VL-72B	23.5% -1.8	78.5% +7.6	0.2% -0.1	23.8% +1.9	75.3% -2.9	1.9% +4.1	23.7% +0.1	76.9% +2.4	1.1% +2.0
Qwen2.5VL-32B	31.7% -4.3	50.0% +22.7	0.0%	26.7% +0.6	70.5% -1.4	0.2% +0.2	29.2% -1.8	60.2% +10.6	0.1% +0.1
Qwen2.5VL-7B	29.0% +11.0	52.2% -11.0	0.6% +21.7	29.3% +6.1	58.7% -3.7	11.4% -4.4	29.2% +8.5	55.4% -7.3	6.0% +8.6
Qwen2.5VL-3B	27.8% +7.7	48.0% -12.1	0.0% +3.3	52.7% -7.3	17.2% +11.2	23.8% -12.8	40.3% +0.2	32.6% -0.4	11.9% -4.7
Qwen3VL-32B	19.7% +1.6	91.7% -0.7	0.0% +0.1	23.4% +0.7	77.9% +0.5	0.1% +0.2	21.5% +1.1	84.8% -0.1	0.04% +0.1
Qwen3VL-8B	19.6% +4.3	80.9% +3.3	0.0% +0.7	25.0% +0.3	69.3% +3.8	0.1% +4.8	22.3% +2.3	75.1% +3.5	0.04% +2.8
Qwen3VL-4B	22.4% +1.6	80.4% +1.3	0.0% +0.9	25.7% -0.9	68.5% +0.9	0.0% +1.1	24.0% +0.3	74.4% +1.1	0.0% +1.0
Qwen3VL-2B	48.4% +2.8	33.6% +11.6	0.1% +13.2	67.4% -17.6	5.6% +32.2	0.0% +10.5	57.9% -7.4	19.6% +21.9	0.04% +11.9
Intern3.5-VL-38B	40.8% +17.6	44.5% -8.9	14.2% -10.3	25.9% -0.1	59.7% +2.0	7.7% -3.3	33.3% +8.7	52.1% -3.4	10.9% -6.8
Intern3.5-VL-14B	34.4% +24.8	52.2% -24.9	8.0% -7.8	30.0% -1.4	53.8% +3.2	0.6% +1.2	32.2% +11.7	53.0% -10.9	4.3% -3.3
Intern3.5-VL-4B	34.7% +15.2	50.3% -20.1	0.0% +0.1	33.6% -4.5	35.3% +16.7	0.9% +0.6	34.2% +5.4	42.8% -1.7	0.5% +0.4
ProgressLM-SFT-3B	19.7%	76.4%	3.2%	32.4%	49.6%	5.8%	26.0%	63.0%	4.5%
ProgressLM-RL-3B	15.5%	88.9%	0.9%	30.9%	46.0%	11.3%	23.2%	67.5%	6.1%

fully stacked. Progress estimation is then performed by comparing subtle state differences, such as the remaining motion of the robot arm and the degree of object contact. This behavior aligns with the strong diagonal structure observed in the Vision Same-View diagnostic heatmaps and the low NSE reported in the corresponding benchmark results, indicating that progress estimation is tightly anchored to the retrieved episodic reference.

Robust reasoning under cross-view variations. The cross-view case presents a more challenging setting, where the observation is captured from a viewpoint different from the demonstration sequence. In the block relocation example, despite significant viewpoint changes, the model successfully retrieves a semantically aligned anchor step representing a late-stage placement of the block. Progress is estimated by reasoning over task-relevant state changes, such as object position and the robot's disengagement from manipulation, rather than relying on pixel-level similarity. The resulting prediction closely matches the ground truth, reflecting the softer but still structured coupling between retrieval and estimation observed in the Vision Cross-View setting. This qualitative behavior is consistent with the broader diagonal patterns and slightly increased NSE seen in cross-view evaluations.

Connection to quantitative trends. These cases help explain the performance gap between vision-based and text-based demonstrations observed across benchmarks. Vision-based demos provide dense and continuous state information, allowing the model to ground episodic retrieval in physical states and perform local mental simulation for progress estimation. This grounding leads to higher PRC and lower AFRR compared to text-based inputs, as confirmed by the quantitative results. Even under viewpoint changes, vision-based demonstrations preserve sufficient semantic structure to support reliable progress reasoning.

Key insight. Together, these visualizations reveal that effective progress estimation relies on retrieving a semantically aligned visual anchor and reasoning locally around that reference. Same-view settings enable near-deterministic coupling between retrieval and estimation, while cross-view settings introduce uncertainty that weakens but does not break this coupling. These qualitative findings provide concrete evidence that vision-based demonstrations play a critical role in enabling robust, generalizable progress reasoning, especially under domain shifts and partial observability.

C.2 Text-Based Demo Case Studies.

We present qualitative case studies to analyze progress estimation with text-based demonstrations. Compared to vision-based demos, text-only instructions provide abstract and discrete descriptions of task execution, requiring the model to ground linguistic steps to visual

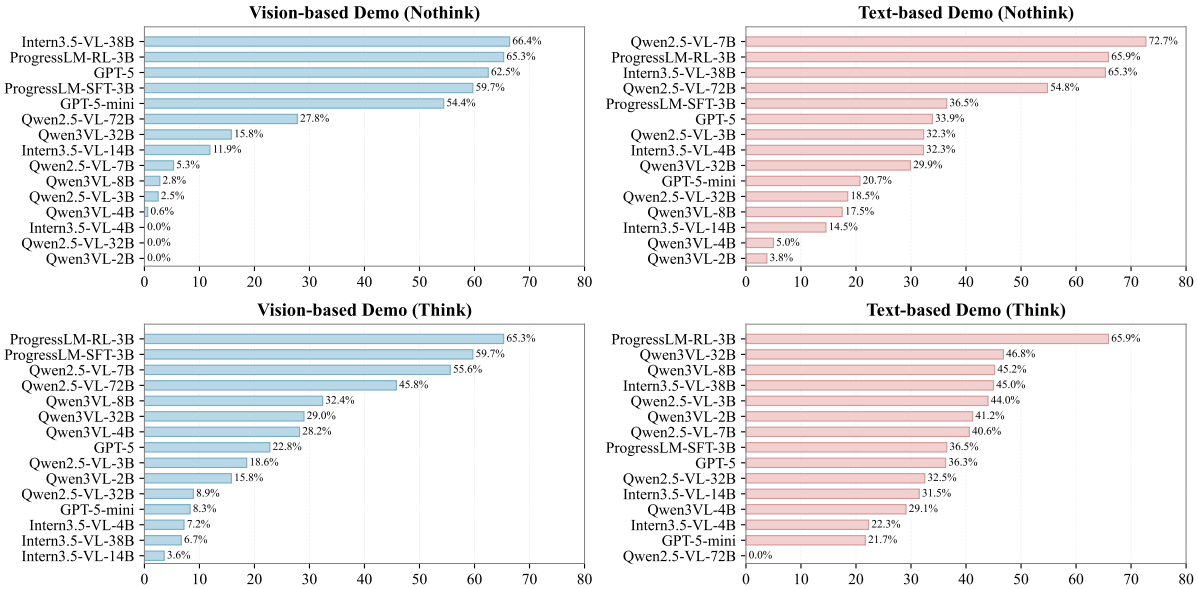


Figure 13: **Unanswerable Detection Accuracy (UDA) across models with and without training-free thinking.** This figure compares unanswerable detection accuracy under Vision-based and Text-based demonstrations, contrasting standard inference (NoThink) with training-free explicit reasoning (Think). Across both modalities, enabling training-free thinking consistently improves UDA for most models, with particularly pronounced gains in text-based settings where semantic mismatch is harder to identify. The results highlight that explicit reasoning at inference time enhances models' ability to recognize ill-defined progress and correctly abstain, complementing the benefits brought by our training-based coupled reasoning approach.

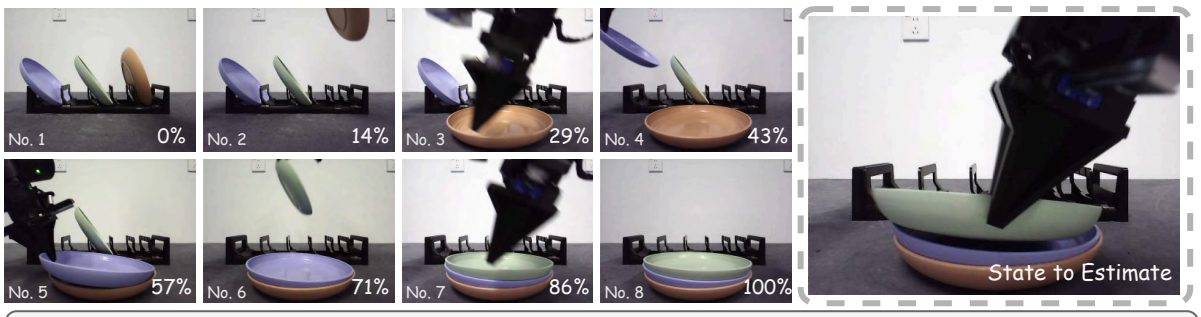
observations before estimating progress.

Episodic retrieval from textual steps. In the cup-stacking example shown in Figure 16, the model first performs episodic retrieval over the textual demonstration and identifies Step 3 as the most semantically aligned reference. This step describes a transitional state where one cup has been placed on the table while another is being lifted. The retrieved anchor reflects a correct alignment between the linguistic action description and the observed physical state, despite the absence of explicit visual cues in the demonstration itself.

Progress estimation under abstract supervision. Conditioned on the retrieved textual anchor, the model estimates progress by reasoning about which sub-actions have been completed and which remain unfinished. In this case, the leftmost cup has been lifted but not yet placed, indicating that the task has just completed Step 3 but has not transitioned to Step 4. Accordingly, the model predicts a progress of 60%, exactly matching the ground truth. This behavior demonstrates that progress estimation is anchored to the relative position within the textual sequence rather than inferred directly from global visual appearance.

Relation to quantitative results. These observations help explain the quantitative trends reported in the main paper. Text-based demonstrations consistently yield higher NSE and AFRR compared to vision-based inputs, reflecting increased ambiguity in episodic retrieval when multiple visual states correspond to the same textual instruction. Nevertheless, the successful alignment in this example shows that when the textual anchor is correctly identified, the model can still perform accurate progress estimation through structured reasoning.

Key insight. These case studies reveal that text-based progress estimation remains feasible but inherently more sensitive to retrieval errors. Effective reasoning depends on accurately grounding abstract textual steps to observed physical states, after which progress estimation can be performed through local comparison within the retrieved episode. This further supports our view that progress esti-



<ref_think>The current state image shows a stack of plates being placed on the table, which is similar to the state in the No. 7 demo image where the plates are also being stacked. The robot arm is still in the process of placing the plates, indicating that the task is almost complete.</ref_think>
<ref>7</ref>
<score_think>Comparing the current state image with the No. 7 demo image, the plates are almost fully stacked, suggesting that the task is nearing completion. The robot arm is still in the process of placing the plates, so the progress is slightly behind the No. 7 demo image.</score_think>
<score>76%</score> (Ground Truth: 80%)

Figure 14: **Vision-Based Case Visualization (Same-View)**. This example illustrates how the model performs progress estimation by coupling episodic retrieval with mental simulation. Given a current observation (right), the model retrieves the most semantically aligned demonstration step (No. 7) from the visual demo sequence (left), where the plates are nearly stacked. Based on this retrieved anchor, the model estimates the relative progress by comparing fine-grained state differences, yielding a progress prediction of 76% against a ground-truth of 80%. The intermediate reasoning explicitly shows how reference selection and score estimation are jointly grounded in the demonstration sequence.

mation, even under purely textual supervision, benefits from an explicit coupling between episodic retrieval and mental simulation.

C.3 Analysis of Coupled Progress Two-Stage Reasoning

To further diagnose whether progress estimation is performed as a coupled reasoning process, we analyze the interaction between **episodic retrieval** and **progress score prediction** at an intermediate level. Specifically, we examine whether the demonstration step selected as the episodic anchor is aligned with the step that is semantically consistent with the predicted progress score.

For each test instance, we record: (1) the **episodic retrieval anchor index**, corresponding to the demonstration step selected by the model as reference; and (2) the **score-aligned demonstration index**, defined as the step whose ground-truth progress interval best matches the model's predicted score. We aggregate these pairs into a 2D histogram, shown as heatmaps (See Figure 12) under two settings: Vision-Based Demonstration (Same-View and Cross-View), and Text-Based Demonstration.

Vision Same-View. Under the vision same-view setting, the heatmap exhibits a clear and sharp diagonal structure, indicating strong alignment between the retrieved anchor and the score-consistent demonstration step. This suggests that when visual states are well aligned, the model reliably retrieves the correct episodic reference and performs progress estimation within its local context. The tight concentration along the diagonal provides strong evidence that progress prediction is not performed as an isolated regression, but is explicitly anchored to episodic retrieval.

Vision Cross-View. In the cross-view setting, the diagonal structure remains evident but becomes noticeably wider, with increased mass in neighboring indices. This reflects higher uncertainty in episodic retrieval caused by viewpoint changes, where multiple demonstration steps may be visually or semantically plausible anchors. Importantly, the distribution still concentrates around the diagonal, indicating that progress estimation remains conditioned on episodic retrieval, albeit in a softer and less deterministic manner.

Text. For text-based demonstrations, the heatmap shows the weakest alignment, with broader dispersion across indices. This behavior is expected, as textual steps often cor-

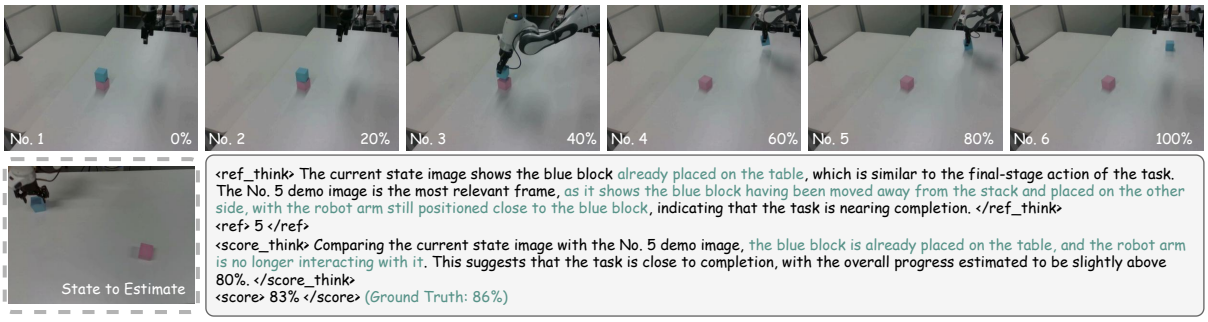


Figure 15: **Vision-Based Case Visualization (Cross-View)**. This example illustrates progress estimation with cross-view visual demonstrations, where the observation is captured from a viewpoint different from the demonstration sequence. Given the current state image (bottom left), the model retrieves the most semantically aligned demonstration step (No. 5), corresponding to a near-completion stage where the blue block has been placed on the table. By comparing fine-grained state differences between the retrieved anchor and the current observation, the model estimates the task progress as 83%, closely matching the ground truth of 86%. This case highlights the model's ability to perform robust episodic retrieval and progress estimation despite viewpoint changes.

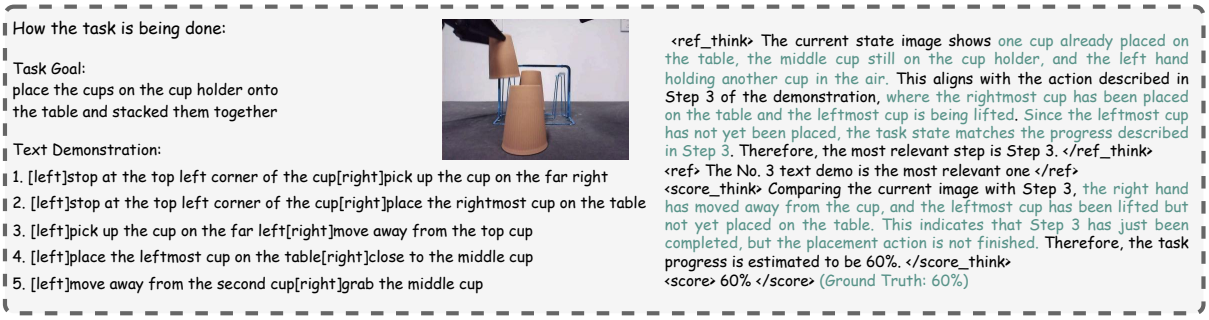


Figure 16: **Text-Based Case Visualization**. This example illustrates progress estimation with text-based demonstrations. Given the current visual observation, the model retrieves the most semantically aligned textual step (Step 3) from the demonstration by matching object states and action semantics described in language. Progress is then estimated by reasoning over the partially completed placement action, resulting in a prediction of 60%, which exactly matches the ground truth. This case highlights how episodic retrieval over textual steps can effectively anchor progress estimation, even when demonstrations are provided purely in language.

respond to abstract or overlapping physical states, making episodic retrieval inherently more ambiguous. Nevertheless, the persistence of a diagonal trend indicates that even in the absence of visual grounding, the model continues to estimate progress relative to a retrieved textual anchor rather than collapsing into direct score regression.

In summary, across all settings, these results consistently demonstrate that progress estimation emerges as a **coupled** reasoning process, where ***episodic retrieval serves as a prerequisite for mental simulation and score estimation***. The gradual degradation from vision same-view to cross-view and text highlights how modality-induced uncertainty weakens but does not remove this coupling, providing mechanistic evidence that supports our model design and training strategy.

C.4 In the Wild Generalization Analysis

We analyze in-the-wild generalization using HUMAN-BENCH, which evaluates progress estimation on human-performed activities under unconstrained settings. Compared to robotic demonstrations, these scenarios introduce substantial domain shifts in embodiment, motion dynamics, object appearance, and execution variability, making reliable progress estimation and abstention behavior significantly more challenging.

Overall trends. As shown in Table 3, most vision-language models exhibit clear degradation in **NSE** and **PRC** under in-the-wild con-

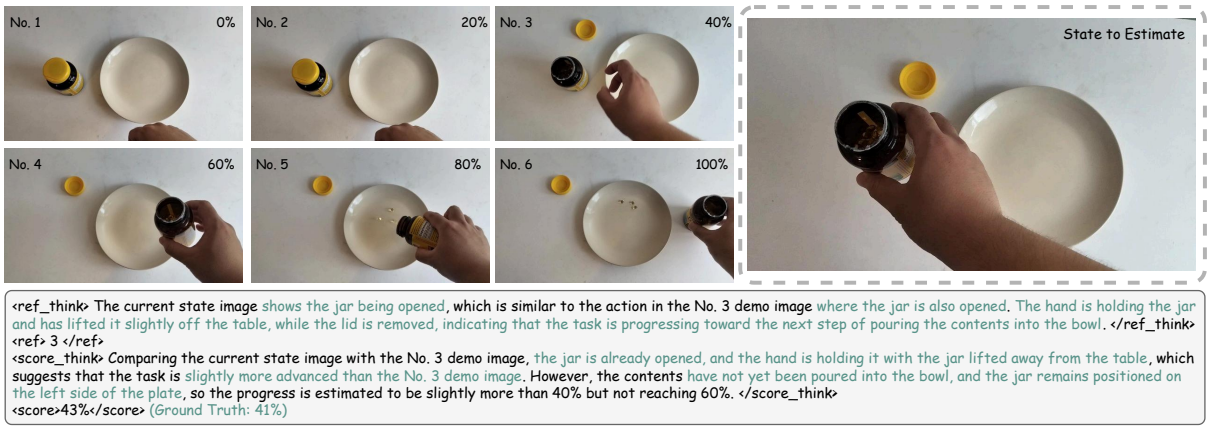


Figure 17: **In-the-wild Generalization on Human Activities**. This example demonstrates the model's ability to generalize coupled progress reasoning beyond robotic manipulation to human-performed activities. Given a sequence of demonstration frames depicting the step-by-step process of opening a jar and pouring its contents, the model retrieves the most semantically aligned demonstration step (No. 3) for the current observation and estimates the task progress by comparing subtle state differences. The predicted progress (43%) closely matches the ground truth (41%), illustrating that episodic retrieval and progress estimation remain effective in unconstrained, real-world human activity scenarios.

ditions, accompanied by elevated **AFRR**. This indicates that human activities amplify both progress miscalibration (higher NSE) and incorrect abstention behavior (higher AFRR), especially when progress must be inferred from subtle hand–object interactions rather than rigid robot motions.

Impact of model scale and visual grounding. Larger models with stronger visual grounding, such as Qwen3VL-32B, consistently achieve higher **PRC** and near-zero **AFRR**, suggesting improved recognition of valid progress states. However, their **NSE** remains relatively high, indicating that increased model capacity alone is insufficient to ensure fine-grained progress calibration in human activities. Smaller variants (e.g., Qwen3VL-2B and Qwen2.5VL-7B) suffer from both elevated NSE and reduced PRC, reflecting compounded errors in episodic alignment and progress estimation.

Vision versus text demonstrations. Across nearly all models, vision-based demonstrations outperform text-based ones in terms of both **NSE** and **PRC**. Text-based inputs consistently yield higher **AFRR**, revealing a tendency to incorrectly abstain when step boundaries are ambiguous. This behavior aligns with the abstract nature of textual steps in human activities, where multiple physical states may correspond to a single instruction, weakening episodic retrieval and downstream progress estimation.

Effectiveness of coupled progress reasoning. Despite its smaller scale, PROGRESSLM-RL-3B achieves the lowest average **NSE** while maintaining competitive **PRC** and controlled **AFRR**. Compared to the SFT-only variant, reinforcement learning consistently reduces NSE, indicating improved calibration of continuous progress estimates. ***These gains suggest that explicitly coupling episodic retrieval with progress estimation is particularly beneficial under domain shift, where robust anchor selection becomes critical***.

Qualitative alignment with in-the-wild cases. The quantitative trends are consistent with the qualitative example in Figure 14. In the jar-opening task, the model retrieves a semantically aligned demonstration step corresponding to the jar being opened and estimates progress by comparing fine-grained state differences, resulting in a prediction (43%) closely matching the ground truth (41%). This example illustrates how accurate episodic anchoring enables stable progress estimation even in the presence of human-specific variability.

Key insight. These results suggest that in-the-wild generalization depends less on raw model capacity and more on whether progress

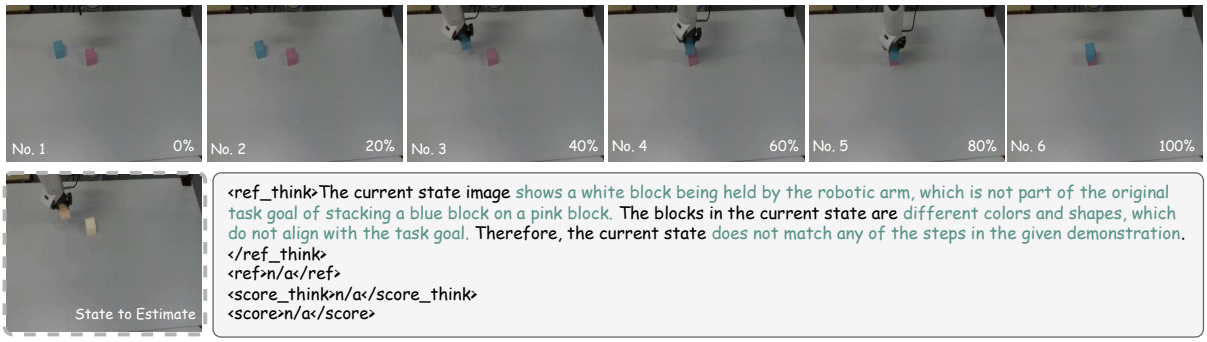


Figure 18: **Visual Unanswerable Case Visualization**. This example illustrates a visual unanswerable scenario where the current observation is semantically inconsistent with the given demonstration. While the demonstration depicts a task of stacking a blue block on a pink block, the observed state shows the robot holding an unrelated white block that does not appear in any demonstration step. As no valid episodic anchor can be retrieved and progress estimation is ill-defined, the model correctly abstains by predicting N/A. This case highlights the model's ability to detect semantic mismatch and avoid spurious progress estimation.

Figure 19: **Text Unanswerable Case Visualization**. This example illustrates a text unanswerable scenario where the current visual observation is semantically incompatible with the textual demonstration. While the task goal and instructions describe stacking bowls on a bowl holder, the observed state contains a stack of cups on the floor, involving different object categories and spatial configurations. As the observation cannot be aligned with any textual step in the demonstration, episodic retrieval fails and progress estimation becomes ill-defined, leading the model to correctly output N/A. This case highlights the model's ability to detect cross-modal semantic mismatch and abstain from spurious progress predictions.

estimation is performed as a structured, coupled reasoning process. Models that fail to anchor progress estimation to semantically aligned episodic references tend to exhibit higher **NSE** and **AFRR**, while PROGRESSLM demonstrates that explicitly modeling this coupling leads to more reliable progress reasoning beyond curated robotic environments.

C.5 Unanswerable Case Recognition

We analyze the ability of models to recognize unanswerable cases, where progress estimation is ill-defined due to semantic mismatch between demonstrations and observations. This capability is critical for safe and reliable progress reasoning, as erroneous score prediction in such cases leads to spurious confidence and degraded downstream performance.

Training-free thinking improves unanswerable recognition. Figure 13 compares unanswerable detection accuracy (UDA) across models under standard inference (*NoThink*) and training-free explicit reasoning (*Think*). A consistent pattern emerges across both vision-based and text-based demonstrations: enabling explicit reasoning at inference time substantially improves UDA for most models. The improvement is especially pronounced in text-based settings, where semantic mismatch is more abstract and harder to detect from surface cues alone.

Vision versus text demonstrations. Under vision-based demonstrations, several large models already achieve moderate UDA in the NoThink setting, suggesting that visual inconsistencies such as object category or spatial violations can often be detected via perceptual cues. However, training-free thinking further improves performance by encouraging explicit comparison between the current state and retrieved demonstration steps, leading to more reliable abstention decisions. In contrast, text-based demonstrations exhibit much lower NoThink performance, indicating that without explicit reasoning, models tend to hallucinate progress scores even when no textual step aligns with the observation. The gains from Think in this setting highlight the importance of structured semantic comparison for detecting cross-modal inconsistency.

Model-dependent effects and limitations of scale. While larger models generally benefit more from training-free thinking, the results reveal that model scale alone does not guarantee robust unanswerable recognition. Several large models still exhibit limited UDA under NoThink inference, particularly for text-based inputs. This suggests that recognizing unanswerable cases requires not only capacity but also an explicit reasoning process that verifies the existence of a valid episodic anchor before attempting progress estimation.

Interaction with training-based coupled reasoning. Notably, models trained with our coupled progress reasoning framework, such as PROGRESSLM, demonstrate strong UDA even without training-free thinking, and further benefit when Think is enabled. This indicates a complementary relationship between training-based supervision and inference-time reasoning: training encourages the model to internalize the prerequisite that progress estimation depends on successful episodic retrieval, while training-free thinking makes this dependency explicit during inference. Together, they lead to more robust detection of ill-defined progress scenarios.

Key insight. These findings suggest that unanswerable case recognition is fundamentally a reasoning problem rather than a purely perceptual one. Reliable detection requires verifying whether a semantically aligned episodic reference exists before estimating progress. Training-free thinking provides a lightweight mechanism to expose this verification process at inference time, while training-based coupled reasoning reinforces it structurally. The combination of both yields the most reliable unanswerable recognition across modalities and model scales.

D Prompts

D.1 Vision-based Demo

D.2 Text-based Demo

D.3 Vision-based Chain-of-Thought Prompt

D.4 Text-based Chain-of-Thought Prompt

D.5 Unanswerable Vision-based Sample Generation

D.6 Unanswerable Text-Based Sample Generation

You are a progress estimator that evaluates the progress of the current state during an ongoing task based on a visual demonstration. The demonstration consists of a sequence of vision-based states and their corresponding progress value (ranging from 0% to 100%), showing how the task evolves from start to completion.

Here is the demonstration:
[Insert the ordered set of demonstration frames, representing sequential progress from earliest stage to latest stage]

Here is the current state that you need to estimate:
[Insert the single image "stage_to_estimate"]

Your task:

1. Check the current state image carefully.
2. Analyze the overall task goal and visual demonstration to understand how the task progresses from start to completion.
3. Identify the reference states from the visual demonstration that are most related to the current state image.
4. Compare the current state image with the chosen reference state, determining whether the image is behind or after the reference state.
5. Estimate the progress numerically as a floating-point value between 0% and 100%.
6. If you really cannot match the current state image to any of the states from demonstration, you need to explain the reason within '<ref_think></ref_think>' and output "n/a" within '<ref></ref>', '<score_think></score_think>', and '<score></score>'.

Your response must strictly follow this format:
`<ref_think>` Reason for choosing the most related state from the demonstration as the reference or explanation of why the current state image does not match the task goal or any steps from demonstration `</ref_think>`
`<ref>` which state from the visual demonstration is most related to the current state (output only the number of the state) or "n/a" `</ref>`
`<score_think>` Reason for comparing the current state image with the reference state or "n/a" `</score_think>`
`<score>` Your final estimated progress score or "n/a" `</score>`

Table 4: Prompt for Visual Demo Inference

You are a progress estimator that evaluates the progress of the current state during an ongoing task based on a textual demonstration. The demonstration consists of a sequence of text-based steps and their corresponding progress value (ranging from 0% to 100%), showing how the task evolves from start to completion.

Here is the demonstration:
[Insert the full ordered text_demo containing all steps and their associated progress values]

Here is the current state that you need to estimate:
[Insert the single image named "stage_to_estimate"]

Your task:

1. Read the task goal to understand the task objective and the entity being operated on.
2. Analyze the textual demonstration to understand how the task progresses from start to completion.
3. Examine the current state image carefully. If the target is incorrect (different from the object metioned in task goal) or you really cannot match the current image to any step in the demonstration, you must explain the reason within `<ref_think></ref_think>` and output "n/a" within `<ref></ref>`, `<score_think></score_think>`, and `<score></score>`.
4. If a match is possible, examine all steps in the textual demonstration, where each step represents an independent action. Identify the single step whose action is most closely related to the current state image. Then compare the current image with that reference step to determine whether it corresponds to an earlier or later stage, and finally estimate the overall progress as a floating-point value between 0% and 100%.

Your response must strictly follow this format:
`<ref_think>` Explain the reason for selecting the most relevant step from the demonstration. If the task target is incorrect, or the current state image cannot be matched to any demonstration step, explain why here. `</ref_think>`
`<ref>` If a valid matching step exists, output only the step number. If the task target is incorrect or no step matches the current image, output only "n/a". Please ensure that this is the same as the ref value you reasoned before. `</ref>`
`<score_think>` If a valid matching step exists, explain how you compare the current image with that step to judge progress. If the task target is incorrect or no step matches the current image, output only "n/a". `</score_think>`
`<score>` If a valid matching step exists, output the estimated progress score (0%–100%). If the task target is incorrect or no step matches the current image, output only "n/a". `</score>`

Table 5: Prompt for Text Demo Inference

You are an expert AI analyst specializing in generating step-by-step reasoning for visual task-progress evaluations. Your objective is not to estimate from scratch. Instead, your task is to construct a perfect, human-like chain of thought that logically explains and justifies a known, ground-truth progress score. Your entire response must read as if you are deducing the conclusion independently from visual analysis alone. You are a progress estimator specializing in evaluating the progress of an ongoing task based on visual evidence. The demonstration consists of a sequence of video frames (images) showing how the task evolves from 0% (start) to 100% (completion). Your goal is to produce a human-like reasoning chain that logically supports the given progress score.

Here is the demonstration:
[Insert the ordered set of demo images representing progress stages, from early to late]

Here is the current state that you need to estimate:
[Insert the single image named "stage_to_estimate"]

Critical Rule
The correct final progress score will be provided to you. However, you must never reveal or imply that you already know the answer. Your reasoning must appear as a fully original, independent visual analysis derived from the images.

Ground-Truth Progress Result
Closest Reference Frame: `{closest_idx_str}`
Final Progress Score to Justify: `{progress_score_str}`

Abnormal Situation Handling:
If you detect any of the following abnormal situations:

- The current state does not match the task goal or any visual demo images
- The operation appears to have failed or resulted in an error state

You must output "n/a" for both `<ref>` and `<score>`. In your reasoning sections, clearly explain why the situation is abnormal and why no valid progress estimation can be made.

Your task:

1. Analyze the demonstration images to understand how the task visually progresses from start to completion.
2. Identify the frame (or frames) from the demonstration that are visually most similar to the current state image.
3. Compare the current state to that reference frame and determine whether it shows more or less progress.
4. Finally, provide a numeric progress estimation between 0% and 100%, or both `<ref>` and `<score>` be "n/a" while encountering abnormal situation.

Your response must strictly follow this format:

`<ref_think>` Your reasoning for choosing the closest demonstration frame as the reference, OR explanation of why the situation is abnormal and no reference can be identified `</ref_think>`

`<ref>` The progress score of your chosen reference frame, OR "n/a" if abnormal situation detected `</ref>`

`<score_think>` Your reasoning for comparing the current state image with the reference frame, OR explanation of why no valid progress score can be assigned `</score_think>`

`<score>` Your final estimated progress score, OR "n/a" if abnormal situation detected `</score>`

Table 6: Chain-of-Thought Construction for Vision-Based Demo

You are an expert AI analyst specializing in visual task-progress evaluations. Your objective is not to estimate from scratch. Instead, your task is to construct a perfect, human-like chain of thought that logically explains and justifies a known, ground-truth progress score. Your entire response must read as if you are deducing the conclusion independently from visual analysis alone.

This is the system prompt for normal inference. You are a progress estimator that evaluates the progress of an ongoing task based on a textual demonstration of its step-by-step progression. The demonstration consists of a sequence of text instructions (text_demo), each describing one step of the process. Each step explicitly states the corresponding progress value (ranging from 0% to 100%), showing how the task evolves from start to completion.

Here is the demonstration:
[Insert the full ordered text_demo containing all steps and their associated progress values]

Here is the current state that you need to estimate:
[Insert the single image named "stage_to_estimate"]

Critical Rule
The correct final progress score will be provided to you. However, you must **never** reveal or imply that you already know the answer. Your reasoning must appear as a fully original, independent visual analysis derived from the images.

Ground-Truth Progress Result
Closest Reference Frame: The No. `{closest_idx}` text demo is the most relevant one
Final Progress Score to Justify: `{final_progress_score}`

Abnormal Situation Handling:
If you detect any of the following abnormal situations:

- The current state does not match the task goal or any demo steps
- The operation appears to have failed or resulted in an error state

You must output "n/a" for both `<ref>` and `<score>`. In your reasoning sections, clearly explain why the situation is abnormal and why no valid progress estimation can be made.

Your task:

1. Analyze the text_demo to understand how the task visually and conceptually progresses from start to completion.
2. Identify the step from the text_demo that are most visually and semantically similar to the current state image.
3. Compare the current state image with the chosen reference step to determine whether it represents an earlier or later stage.
4. Estimate the progress numerically as a floating-point value between 0% and 100%, or both `<ref>` and `<score>` be "n/a" while encontering abnormal situation.

Your response must strictly follow this format:

`<ref_think>` Your reasoning for choosing the most similar text_demo step as the reference, OR explanation of why the situation is abnormal and no reference can be identified `</ref_think>`

`<ref>` which text demo is most semantically similar to the current state (output only the number), OR "n/a" if abnormal situation detected `</ref>`

`<score_think>` Your reasoning for comparing the current state image with the reference step, OR explanation of why no valid progress score can be assigned `</score_think>`

`<score>` Your final estimated progress score, OR "n/a" if abnormal situation detected `</score>`

Table 7: Chain-of-Thought Construction for Text-Based Demo

You are tasked with constructing adversarial image edits that intentionally cause failure in an instruction-following, multi-step visual manipulation task while preserving realism and coherence. Your goal is to make the provided image no longer align with its corresponding step instruction.

Input Information:
Task Goal: `{task_goal}`
Step-by-step Instructions: `{text_demo}`
Current Image: [The provided image]
Corresponding Instruction: Step `{step_number}` – specific_instruction

Your Task:
You are given an image that corresponds to a specific step in a multi-step robotic manipulation task. Your goal is to edit this image to make it no longer align with the corresponding instruction, causing the task to fail.

Editing Guidelines:
Modify key objects or elements in the image using one of the following strategies:

1. Color Change: Alter the color of critical objects (e.g., change a red apple to green)
2. Object Replacement: Replace the target object with a different object (e.g., replace an egg with an orange)
3. Occlusion/Removal: Hide or remove key objects from the scene

Requirements:

1. The edited image should clearly violate the corresponding instruction.
2. Maintain visual realism and coherence—the edited image must look natural and believable.
3. Ensure the edit would cause the overall task goal to fail.
4. The modification should be semantically meaningful (not just noise or blur).

Output Format:

`<strategy_think>` Analyze the current instruction and image content. Think step by step about which editing strategy would most effectively violate this instruction while maintaining realism. Consider the key objects involved and how modifying them would break the instruction. `</strategy_think>`

`<strategy>` State the single strategy you selected from the editing guidelines (e.g., "Object Replacement" or "Color Change") `</strategy>`

`<prompt_think>` Think step by step about how to formulate a clear and effective image editing prompt. Consider: What specific change to make? Which objects to target? What details are needed for realism? `</prompt_think>`

`<prompt>` Write a concise image editing prompt (maximum 20 words) that clearly instructs the editing model what to change in the image. `</prompt>`

Table 8: Adversarial Image Editing Prompt Generation for Unanswerable Visual Samples

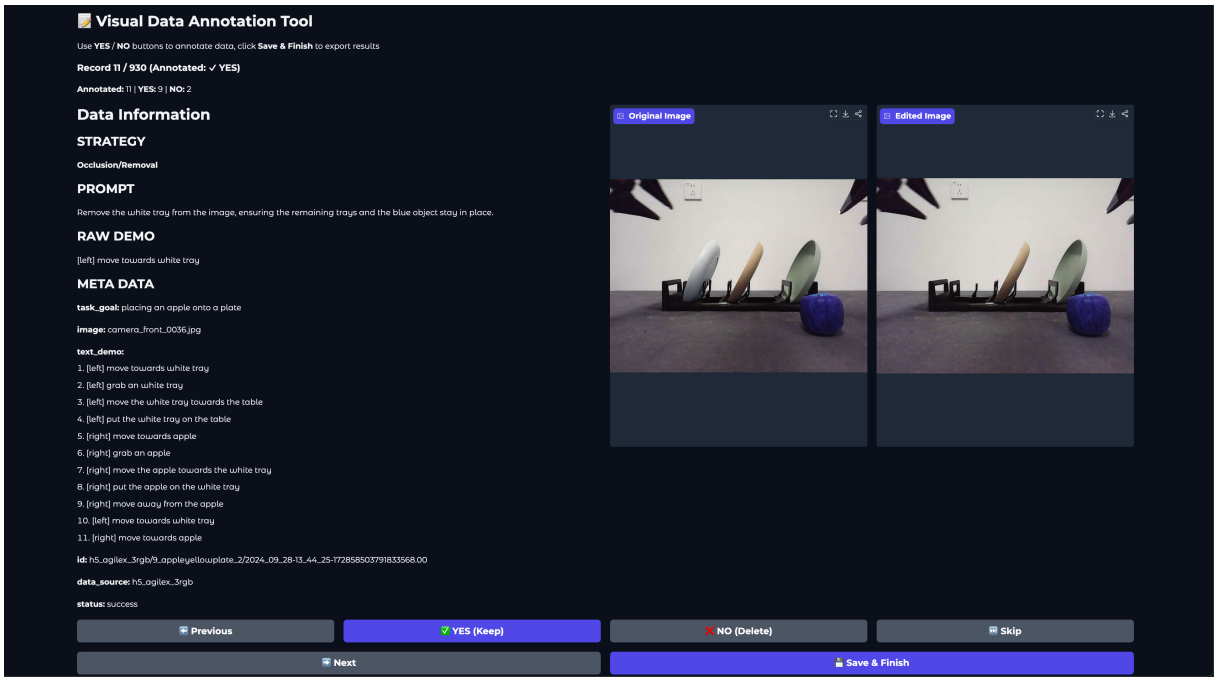


Figure 20: **Gradio-based Human Filtering Platform for Visual Unanswerable Data Generation.** We employ a Gradio-based annotation interface to manually verify the quality of edited images used for visual unanswerable construction. Annotators are presented with the original and edited images alongside the task goal, step-level demonstrations, editing strategy, and prompt. Each edited sample is retained only if it simultaneously violates the intended manipulation step and preserves visual realism, ensuring high-quality and reliable visual unanswerable data.

Task: Task: Modify the Task Goal and Step-by-step Instructions to make the Current Image does not match the Task Goal or any Step-by-step Instructions.

Input Information:
- Task Goal: task_goal
- Step-by-step Instructions: text_demo
- Current Image: [The provided image]

Editing Guidelines:
1. Keep the original sentence format and structure - ONLY replace the object name.
2. For each step in Step-by-step Instructions, preserve ALL markers like [right], [left], [towards], etc. in their EXACT original positions.

Output Format:
<edited_goal> "put your edited task goal here" </edited_goal>
<edited_demo>
"text_demo": ["your edited step 1", "your edited step 2", "your edited step 3", ..., "your edited step n"]
</edited_demo>

Table 9: Object Replacement for Unanswerable Text Sample Generation