



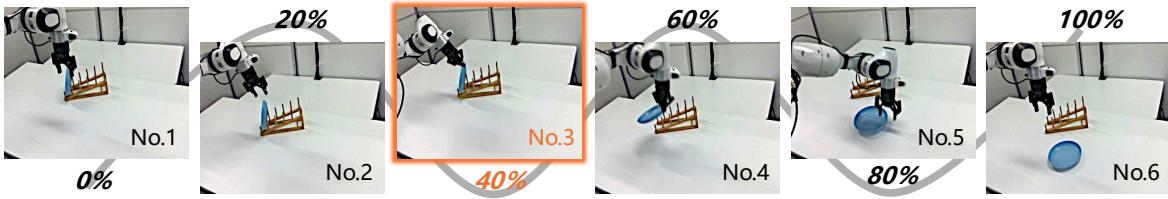
PROGRESSLM: Towards Progress Reasoning in Vision-Language Models

Jianshu Zhang^{1*} Chengxuan Qian^{2*} Haosen Sun¹
Haoran Lu¹ Dingcheng Wang¹ Letian Xue¹ Han Liu¹

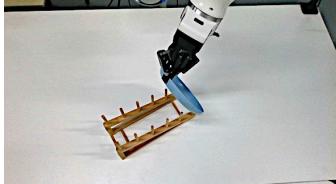
¹Northwestern University ²Arcadia University

Code Model Dataset

First see how the task is being done:



Then, tell me how much of this task has been completed so far?



Direct Prediction
The task is not finished yet. The task is still in progress. About 70%.

Progress Reasoning

<ref_think> The current state shows a plate being lifted from a rack, which is most related to the No. 3 demo image where the plate is also being lifted. The robot's arm is extended, and the plate is in the air. This suggests that the current state is closely related to the No. 3 demo image. <ref_think> <ref> 3 </ref>
<score_think> Comparing the current state image with the No. 3 demo image, the current state image shows the plate slightly higher, indicating that the task is slightly more advanced. Therefore, the progress is estimated to be slightly more than the 40% progress of the No. 3 image. </score_think> <score> 46% </score>

Episodic Retrieval



Mental Simulation

Figure 1: Given a task demonstration and a single observation, the goal is to estimate *how much of the task has already been completed*. 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, enabling a fine-grained estimate of *completed progress*, which enables accurate and interpretable progress estimation.

Abstract

Estimating task progress requires reasoning over long-horizon dynamics rather than recognizing static visual content. While modern Vision-Language Models (VLMs) excel at describing what is visible, it remains unclear whether they can infer *how far a task has progressed* from partial observations. In this work, we introduce PROGRESS-BENCH, a benchmark for systematically evaluating progress reasoning in VLMs. We further explore a human-inspired two-stage progress reasoning paradigm that combines episodic retrieval with mental simulation. We instantiate this paradigm through both training-free prompting and a training-based approach built on an automatically curated dataset, PROGRESSLM-45K. Evaluating 14 VLMs on PROGRESS-BENCH, we find that current models struggle to reliably estimate task progress. While training-free prompting that enforces structured progress reasoning yields improvements, these gains are limited and model-dependent. In contrast, PROGRESSLM-3B achieves consistent improvements in accuracy, robustness to viewpoint variation, and calibrated handling of unanswerable cases, even at small model scale. Further analyses reveal characteristic error patterns of existing VLMs and clarify when and why progress reasoning succeeds or fails.

* Equal Contribution

1 Introduction

Given an observation captured at a particular moment during task execution, most Vision Language Models (VLMs) (Hurst et al., 2024; Bai et al., 2025; Wang et al., 2025) are highly capable of answering questions such as “What is shown in this image?”. However, if the question is instead framed as “How much of this task has been completed so far?”, the problem becomes fundamentally different: it pushes the model to reason about the world from a long-horizon, dynamic perspective, rather than focusing solely on what is immediately visible.

Prior work on progress estimation either relies on task-specific regression models (Yang et al., 2024; Chen et al., 2025), or infers progress indirectly by reformulating the problem into surrogate objectives such as shuffle-and-rank (Ma et al., 2024b) or pairwise comparison (Zhai et al., 2025). This naturally raises the question: *can vision–language models acquire progress estimation as a general reasoning capability from a single observation?* To systematically study this question, we introduce **PROGRESS-BENCH**. We select robotic manipulation tasks (Wu et al., 2025b) as a controlled and representative domain, where task execution exhibits clear, interpretable, and temporally ordered progressions. Each instance provides a task *demonstration* and a single *observation*, and the model is required to predict a normalized *progress score* indicating how far the task has progressed.

PROGRESS-BENCH is constructed along three key dimensions. **(i) Demonstration modality** compares vision-based demonstrations that present state trajectories with text-based demonstrations that provide step-by-step action descriptions. **(ii) Viewpoint correspondence** controls whether demonstrations and observations are captured from the same camera viewpoint or from different viewpoints. **(iii) Answerability** explicitly distinguishes between cases where progress is well-defined and cases where reliable estimation is inherently ambiguous. This design allows us to disentangle perception, temporal reasoning, and uncertainty awareness in progress estimation.

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-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.

In summary, our main contributions are as follows:

1. We introduce PROGRESS-BENCH, comprising over 3K progress estimation instances, to systematically evaluate whether VLMs can perform progress reasoning from a single observation under controlled variations of demonstration modality, viewpoint, and answerability.
2. We conduct a comprehensive evaluation of 14 VLMs on PROGRESS-BENCH and show that current models exhibit limited and unstable progress reasoning ability. They are highly

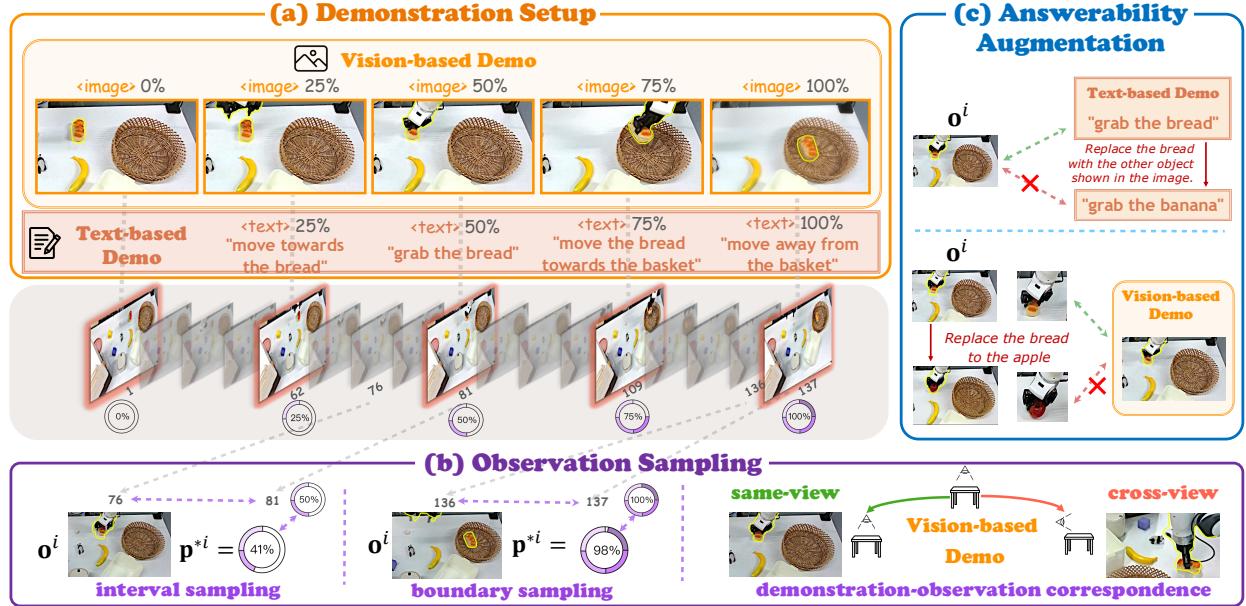


Figure 2: Overview of PROGRESS-BENCH construction. (a) **Demonstration setup**: tasks are presented as either vision-based demonstrations with key frames or text-based ones with step-wise actions, each annotated with progress scores. (b) **Observation sampling**: observations are sampled from intermediate or boundary positions between demonstration steps, with progress labels assigned by interpolation; vision-based settings further include same-view and cross-view demonstration–observation correspondence. (c) **Answerability augmentation**: unanswerable cases are created by introducing semantic mismatches between demonstrations and observations.

sensitive to demonstration modality and viewpoint changes, frequently fail to recognize unanswerable cases, and often collapse progress estimation to coarse or heuristic predictions. Further analyses reveal systematic error patterns underlying these failures.

3. We further explore *how progress reasoning can be improved* by applying a human-inspired reasoning paradigm. Through both training-free and training-based learning, we show that while training-free reasoning yields only conditional benefits, explicit training leads to PROGRESSLM-3B, which achieves performance comparable to or exceeding GPT-5 on PROGRESS-BENCH.

2 PROGRESS-BENCH

Overview. PROGRESS-BENCH evaluates whether a model can situate a single observation within the temporal structure of an ongoing task, going beyond static perception to reason about task progression. Rather than introducing all design factors at once, we organize the benchmark construction into successive stages. Specifically, we vary *demonstration modality* during demonstration setup to control how task execution is presented, introduce *viewpoint correspondence* during observation sampling while progress labels naturally arise from temporal interpolation, and finally introduce an explicit *answerability* dimension to distinguish well-defined progress estimation from inherently ambiguous cases.

2.1 Problem Formulation

Each instance consists of a task demonstration D and an observation o sampled from an intermediate moment of task execution. The demonstration describes the task from start to completion and is presented either as a sequence of images or as a stepwise textual actions. Given (D, o) , the model predicts a progress score $p \in [0, 100\%]$. When the demonstration is insufficient or inconsistent for meaningful estimation for the current observation, the model should output N/A.

2.2 Benchmark Construction

Demonstration Setup. We consider two complementary demonstration modalities. In the *vision-based* setting, demonstrations are sequences 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 world state at a key step. In the *text-based* setting, demonstrations provide action-only step descriptions: $\mathcal{D}_t = \{\mathbf{t}_j, \mathbf{p}_j\}_{j=1}^N$. Compared to vision-based demonstrations, text-based ones require integrating action semantics and reasoning about cumulative effects, as intermediate states are not directly observable.

Observation Sampling. Observation sampling introduces both progress annotation and viewpoint correspondence. Given \mathcal{D} and its raw execution video $\mathbf{V} = \{\mathbf{f}_k\}_{k=1}^T$, we construct observation-progress pairs $\mathcal{O} = \{(\mathbf{o}^i, \mathbf{p}^i)\}_{i=1}^M$. A demonstration partitions the video into $N-1$ temporal segments between consecutive key steps. For each segment, we sample one or a few intermediate frames by choosing a relative position $\delta \in (0, 1)$, yielding an observation \mathbf{o} . We assign progress by linear interpolation,

$$\mathbf{p}^* = \mathbf{p}_j + \delta(\mathbf{p}_{j+1} - \mathbf{p}_j), \quad \delta \in (0, 1), \quad (1)$$

which is justified by smoothly executed expert videos where progress advances continuously and approximately monotonically between consecutive key steps.

To balance coverage and granularity, we consider two choices of δ :

$$\delta \in \begin{cases} \left\{ \frac{1}{K}, \dots, \frac{K-1}{K} \right\}, & (\text{interval sampling}), \\ (1 - \epsilon, 1), & (\text{boundary sampling}). \end{cases} \quad (2)$$

To test viewpoint robustness, we further control demonstration-observation correspondence in the vision-based setting. In *same-view* settings, demonstrations and observations are captured from the same camera pose; in *cross-view* settings, observations come from a different viewpoint. This discourages shortcut matching on pixel similarity and probes whether models can reason about progression under viewpoint changes.

Answerability Augmentation. Progress understanding requires not only accurate predictions when progress is well-defined, but also appropriate abstention when it is not. We therefore augment PROGRESS-BENCH with an *answerability* dimension. *Answerable* samples correspond to coherent executions where progress can be meaningfully inferred. *Unanswerable* samples arise when the observation is inconsistent with the demonstration, rendering progress estimation ill-defined; the correct output is N/A. We construct such mismatches via two complementary mechanisms: (i) modifying the textual demonstration while keeping the observation fixed, and (ii) editing the observation image (with an image-editing model) while preserving the original demonstration.

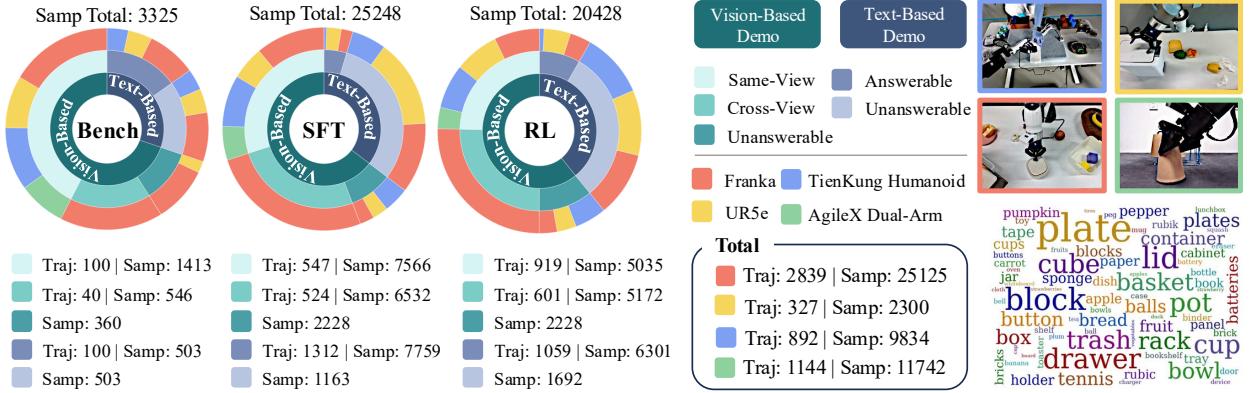


Figure 3: Data statistics of PROGRESS-BENCH and PROGRESSLM-45K (25K for SFT while 20K for RL). Traj and Samp denote the numbers of task trajectories and sampled observations to be estimated, respectively. The upper-right panel shows the four distinct robotic embodiments included, while the lower-right panel visualizes the diversity of objects involved in task interactions.

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 which corresponds to 3325 questions.

3 Towards Progress Reasoning in VLMs

We conceptualize progress reasoning as a two-stage process inspired by how humans judge task progress from partial observations. Given a demonstration and a current observation, humans first perform *episodic retrieval*, recalling a representative moment from the demonstration that most closely resembles the current state and using it as a coarse anchor. They then conduct *mental simulation*, reasoning about how the task would evolve from this anchor to the present observation to estimate relative progress. This formulation treats progress estimation as reasoning over a latent task trajectory, rather than matching observations to precise timestamps.

3.1 Training-Free Approach

We instantiate the proposed two-stage progress reasoning via structured prompting, without any parameter updates. The prompt enforces an explicit output schema that decomposes reasoning into four fields: `<ref_think>` for episodic retrieval reasoning, `<ref>` for the retrieved reference step, `<score_think>` for mental simulation comparing the reference with the current observation, and `<score>` for the final progress estimate. At inference time, the model is instructed to generate outputs following this format.

3.2 Training-Based Approach

We also adopt a training-based approach that explicitly teaches *episodic retrieval* and *mental simulation*. We construct PROGRESSLM-45K, using manipulation tasks that never appear in PROGRESS-BENCH, preventing task memorization and enforcing transferable reasoning. Data statistics and composition are summarized in Figure 3.

Cold-Start Supervised Fine-Tuning. The purpose of the cold-start stage is to help the model internalize the two-stage reasoning pattern of *episodic retrieval* followed by *mental simulation*. To this end, we construct PROGRESSLM-25K-COT, where each training instance is of the form $\mathcal{D}_{\text{train}} = \{(\mathcal{D}^i, \mathbf{o}^i, \mathbf{r}^{i*})\}$. The reasoning sequence \mathbf{r}^{i*} contains both the final progress prediction (`<score>`) and a ground-truth reference step (`<ref>`). To generate intermediate reasoning traces, we adopt a *guided reasoning completion* strategy, where the model is provided with ground-truth `<ref>` and `<score>` and is required to generate the missing reasoning fields `<ref_think>` and `<score_think>`. We then perform supervised fine-tuning by minimizing the autoregressive negative log-likelihood:

$$\mathcal{L}_{\text{SFT}} = -\frac{1}{N} \sum_{i=1}^N \log P_\theta(\mathbf{r}^{i*} | \mathcal{D}^i, \mathbf{o}^i). \quad (3)$$

Reinforcement Learning. While cold-start fine-tuning teaches the model *how* to reason, it does not ensure robust or well-calibrated predictions. We therefore apply a second-stage reinforcement learning procedure based on GRPO (Shao et al., 2024), with the objective

$$\mathcal{L}_{\text{RL}} = -\mathbb{E}_{\mathbf{r} \sim P_\theta(\mathbf{r} | \mathcal{D}, \mathbf{o})} [\alpha R_{\text{format}} + \beta R_{\text{ref}} + \gamma R_{\text{score}}], \quad (4)$$

where the rewards respectively encourage structured outputs, accurate reference retrieval, and precise progress estimation. Since reasoning is generated autoregressively, accurate episodic retrieval provides a stronger context for subsequent mental simulation, leading to improved overall performance. We use 20K samples in this stage and set $\alpha:\beta:\gamma = 1:6:3$ in our implementation.

4 Evaluation on PROGRESS-BENCH

Experimental Setup. We evaluate 14 VLMs on PROGRESS-BENCH, including GPT-5 and GPT-5-mini (OpenAI, 2024), as well as models from the Qwen (Bai et al., 2025) and Intern families (Wang et al., 2025), with parameter sizes ranging from 2B to 72B. For each model, we consider three evaluation settings. **(i) Direct prediction:** the model is only asked to output a progress score. **(ii) Training-free approach:** the model is prompted with our human-inspired progress reasoning format to produce the final prediction. **(iii) Training-based approach:** based on Qwen2.5-VL-3B, we get PROGRESSLM-SFT and PROGRESSLM-RL. Importantly, the tasks during training have no overlap with those in PROGRESS-BENCH, ensuring that all results reflect generalization to unseen tasks rather than memorization. Detailed experimental settings are provided in Appendix B.

Evaluation Design. Progress estimation involves both local numerical accuracy and global temporal consistency, and some observations are inherently unanswerable due to insufficient visual evidence. We therefore report four complementary metrics to capture pointwise error, trajectory-wise ordering consistency, and answerability awareness. Specifically, (1) **Normalized Score Error (NSE)** measures single-point progress accuracy for answerable samples: given a ground-truth progress score $g \in [0, 100\%]$ and a predicted score $\hat{p} \in [0, 100\%]$, the error is defined as $\text{NSE} = |\hat{p} - g| / \max(g, 1 - g)$, and the dataset score is computed by averaging NSE over all valid answerable samples. (2) **Progress Rank Correlation (PRC)** measures trajectory-level temporal consistency. For a trajectory τ with valid sample indices \mathcal{V}_τ , PRC is defined as the Spearman rank correlation $\text{PRC}(\tau) = \rho_s(\{\hat{p}_i\}_{i \in \mathcal{V}_\tau}, \{g_i\}_{i \in \mathcal{V}_\tau})$, where $\{\hat{p}_i\}$ and $\{g_i\}$ denote the sequences of predicted and ground-truth progress scores along the trajectory, respectively. The final score is obtained by averaging over all valid trajectories (Spearman, 1961). Values closer to 100% indicate more

Table 1: Performance comparison on *answerable* samples under vision-based and text-based demonstrations. We report Normalized Score Error (NSE)↓, Progress Rank Correlation (PRC)↑, and Answerable False Rejection Rate (AFRR)↓, with micro and macro averages. Best, Second Best, and Third Best results are highlighted. Colored deltas show the effect of training-free progress reasoning relative to direct prediction (green: improvement; red: degradation).

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

accurate preservation of the relative temporal ordering of task progress. (3) **Answerable False Rejection Rate (AFRR)** is defined as the fraction of answerable samples predicted as unanswerable. (4) **Unanswerable Detection Accuracy (UDA)** is defined as the fraction of unanswerable samples correctly predicted as unanswerable.

4.1 Performance on Answerable Scenarios

We first analyze model performance on *answerable* samples, where task progress is well-defined and can be inferred from the observation. Table 1 summarizes results under both vision-based and text-based demonstration settings.

How well do current VLMs perform at progress estimation? *Overall, current VLMs exhibit limited and highly variable progress estimation ability under direct prediction.* While strong models such as GPT-5 and Qwen2.5-VL-72B achieve relatively better performance, they remain highly sensitive to the demonstration modality, with vision-based demonstrations consistently outperforming text-based ones. The causes of this modality gap are examined in detail in the subsequent analysis (Section 5.4). We further observe abnormally low, negative, or even undefined PRC values for several models, indicating severely distorted or collapsed progress rankings rather than meaningful ordinal reasoning. As discussed later in Section 5.1, these failures are primarily driven by degenerate predicted score distributions, such as collapse to extreme or discrete values. Finally, some models (e.g., Intern3.5-VL-38B) show pronounced failure modes on answerable samples, with extremely high Answerable False Rejection Rates (AFRR). Rather than reflecting true uncertainty, this behavior suggests an overly conservative strategy that rejects many valid cases where progress can in fact be inferred.

Does training-free progress reasoning help? *Training-free, human-inspired reasoning only benefits large models and harm small models.* For large-scale models (e.g., GPT-5 and Qwen2.5-

Table 2: Performance under *same-view* and *cross-view* observation settings on answerable vision-based demonstrations. We report NSE \downarrow , PRC \uparrow , and AFRR \downarrow . Best, Second Best, and Third Best results are highlighted. Colored deltas indicate the effect of training-free progress reasoning relative to direct prediction (green: improvement; red: degradation).

Model	Same-View			Cross-View			Delta (Same \rightarrow Cross)		
	NSE \downarrow	PRC \uparrow	AFRR \downarrow	NSE \downarrow	PRC \uparrow	AFRR \downarrow	Δ NSE \downarrow	Δ PRC \uparrow	Δ AFRR \downarrow
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

VL-72B/32B), prompting-based reasoning yields consistent gains in PRC and often reduces NSE, suggesting improved both sample-wise and trajectory-wise progress estimation performance. However, smaller models frequently show marginal or negative effects, particularly increased NSE or AFRR, indicating that limited-capacity models may imitate the reasoning format without genuinely improving progress understanding.

Does training-based progress reasoning help? *Explicit training leads to consistent and substantial improvements, even at small model scale.* PROGRESSLM outperforms the base 3B model across all answerable metrics, with PROGRESSLM-RL-3B achieving the best macro-averaged NSE and PRC among all evaluated models. Notably, these gains are achieved despite the much smaller model size, demonstrating that robust progress reasoning is not merely a consequence of scale, but can be effectively learned through targeted supervision and optimization.

4.2 Performance under Viewpoint Variation

To analyze robustness under viewpoint changes, we further decompose the vision-based results in Table 1 into *same-view* and *cross-view* settings in Table 2, where the demonstration and observation are captured from the same or different camera viewpoints.

How do current VLMs handle viewpoint changes? *Most VLMs suffer a clear performance drop under cross-view observations.* Compared to same-view settings, cross-view cases consistently exhibit higher Normalized Score Error (NSE) and lower Progress Rank Correlation (PRC). This degradation is especially pronounced for small and medium-sized models, suggesting a strong reliance on low-level visual similarity rather than viewpoint-invariant progress reasoning.

Does progress reasoning improve robustness under viewpoint variation? *Training-free progress reasoning offers limited and conditional robustness gains.* As shown in Table 2, its effective-

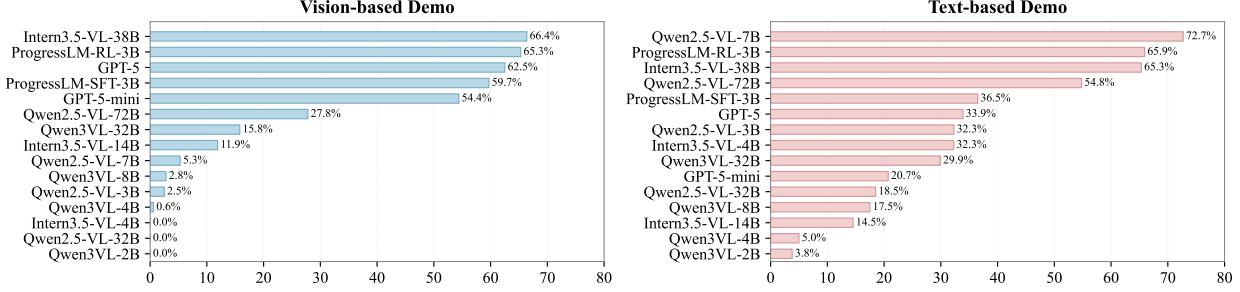


Figure 4: Unanswerable Detection Accuracy (UDA) across models under two settings.

ness depends strongly on the model’s baseline capability: weaker models see little benefit, while strong models such as GPT-5 exhibit only modest cross-view gains, often accompanied by drops in same-view performance, indicating a ceiling effect. When improvements do occur, they are primarily reflected in PRC rather than NSE, suggesting that training-free reasoning mainly preserves relative temporal ordering across viewpoints rather than improving pointwise accuracy. *In contrast, robust cross-view progress reasoning emerges only when the reasoning process is explicitly learned.* PROGRESSLM-3B-RL exhibits consistently smaller same-view to cross-view gaps, indicating improved generalization beyond surface-level visual correspondence.

4.3 Performance on Unanswerable Scenarios

We next evaluate model behavior on *unanswerable* samples, where the observation does not provide sufficient information to reliably infer task progress. This section reports the accuracy of unanswerable case recognition under vision-based and text-based demonstrations.

Can models handle unanswerable scenarios appropriately? *Most VLMs fail to reliably recognize when progress estimation is not possible.* As shown in Figure 6, the majority of evaluated models tend to produce a progress score even when the observation lacks sufficient information, indicating limited awareness of unanswerable scenarios. This behavior suggests that unanswerability is often ignored rather than explicitly recognized. *In contrast, PROGRESSLM consistently identifies unanswerable cases under both vision-based and text-based demonstrations instead of forcing arbitrary predictions.* This capability is further strengthened by reinforcement learning, with PROGRESSLM-3B-RL achieving the strongest or near-strongest unanswerable recognition accuracy across settings. However, strong unanswerable detection alone is not sufficient. As a representative counterexample, INTERNVL-3.5-38B attains relatively high UDA, but Table 1 shows that this comes at the cost of an extremely high Answerable False Rejection Rate (AFRR), reflecting a broadly conservative behavior that fails to distinguish ambiguous cases from answerable ones. Overall, these results highlight that effective progress reasoning requires *selective* uncertainty recognition, rather than indiscriminate rejection or blind overconfidence.

5 Further Analysis

5.1 Distribution of Predicted Score Analysis

To probe how models internally represent task progress beyond aggregate metrics, we analyze the distribution of predicted progress scores across models and input settings (Figure 5).

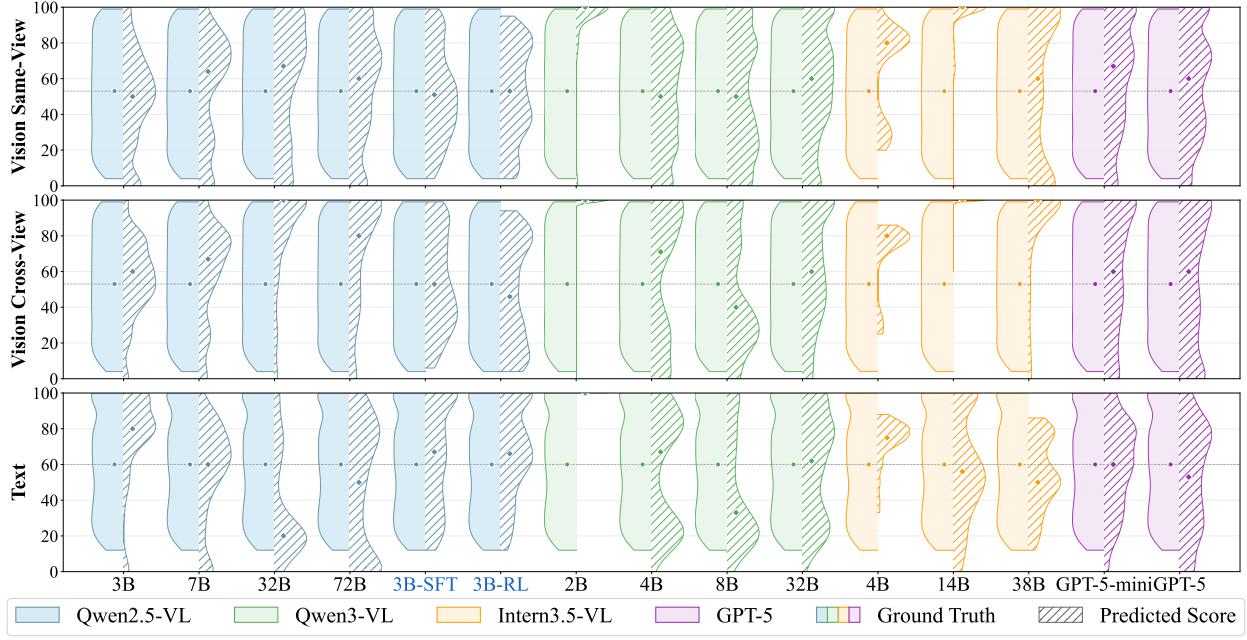


Figure 5: Distribution of predicted progress scores. Some models exhibit collapsed or clustered distributions at extreme or discrete values, indicating reliance on heuristic anchors rather than continuous progress modeling. In contrast, GPT-5 and PROGRESSLM (3B-SFT and 3B-RL) produce smoother distributions, reflecting improved sensitivity to intermediate task progress.

What patterns emerge in predicted progress score distributions? Across models and settings, predicted progress scores often cluster around specific values rather than varying smoothly. We identify four characteristic patterns: *(i) Single-peak collapse*, where predictions concentrate at one value (e.g., 0% or 100%), reducing progress estimation to a near-discrete decision; *(ii) Multi-peak clustering*, where scores alternate among a small set of heuristic anchors; *(iii) Central-peaked distributions*, where predictions accumulate around a mid-range value (typically $\sim 50\%$), reflecting default or uncertain responses; and *(iv) Smooth continuous distributions*, where scores spread broadly over [0, 100%], indicating sensitivity to intermediate task states. Notably, PROGRESSLM-3B-SFT and PROGRESSLM-3B-RL consistently exhibit the fourth pattern across settings, while most base VLMs fall into the first three. These distributional differences explain the distorted or unstable rank correlations observed for many models, and highlight that fine-grained progress reasoning emerges only when progress is explicitly learned.

5.2 Distribution of Per-sample Error Analysis

To assess model reliability beyond aggregate metrics, we analyze the distribution of per-sample prediction errors, as shown in Figure 6.

How do error distributions reflect robustness in progress estimation? *Smaller models are error-unstable, while explicit progress reasoning substantially improves per-sample robustness.* Smaller models exhibit broad, heavy-tailed error distributions, indicating unstable progress estimation with frequent large errors. This pattern is consistent with the degenerate and spiky score distributions observed earlier, where coarse or discretized progress outputs amplify pointwise error. In contrast, larger models produce more compact error distributions with errors concentrated closer to zero.

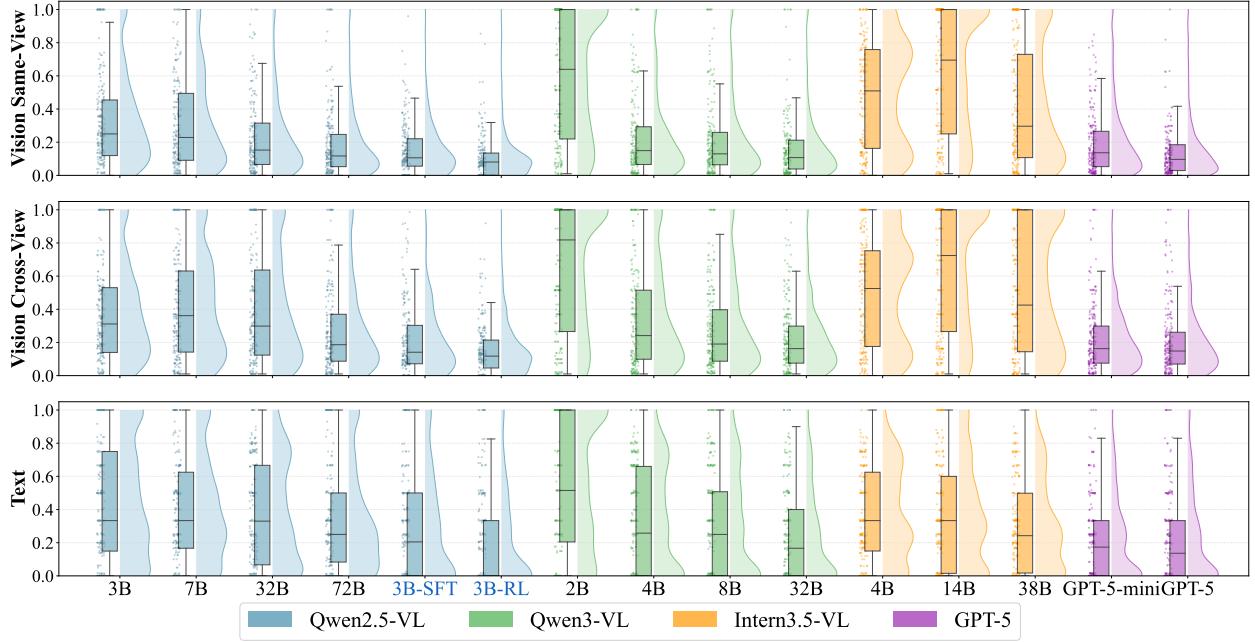


Figure 6: Raincloud plots of per-sample normalized score prediction error across models and settings. Each plot combines jittered samples, box plots, and kernel density estimates. Smaller models exhibit highly dispersed and heavy-tailed error distributions, while larger and our models show more concentrated errors with fewer extreme outliers.

Notably, both PROGRESSLM-3B-SFT and PROGRESSLM-3B-RL significantly tighten the error distribution compared to the base 3B model across all settings. In particular, PROGRESSLM-3B-RL effectively suppresses extreme-error cases, demonstrating that explicitly learned progress reasoning improves not only average accuracy but also robustness at the individual-sample level.

5.3 Coupled Two-Stage Progress Reasoning

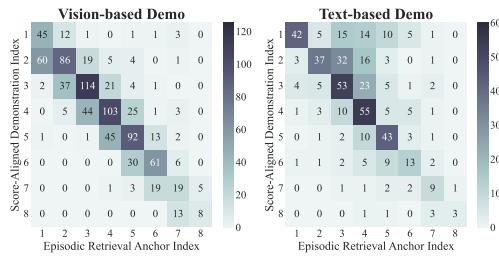


Figure 7: Coupling between the two stages progress reasoning of PROGRESSLM. A diagonal concentration indicates that the anchor selected during *episodic retrieval* consistently constrains second-stage *mental simulation*.

Are the two reasoning stages coupled rather than independent? Yes—the anchor retrieved in the first stage systematically constrains subsequent progress estimation. To examine this coupling, we analyze the relationship in Figure 7 between the *Episodic Retrieval Anchor Index* and the *Score-Aligned Demonstration Index*. The former denotes the demonstration step retrieved during first-stage *episodic retrieval*, while the latter corresponds to the demonstration step whose annotated progress is closest to the model’s final predicted score. If the two stages are truly coupled, these two indices should closely align, producing a near-diagonal distribution. The clear diagonal concentration of our training-based model indicates that anchor retrieval is not an auxiliary signal, but directly guides fine-grained progress estimation through second-stage *mental simulation*.

5.4 Implicit State Accumulation in Text-Based Demonstrations

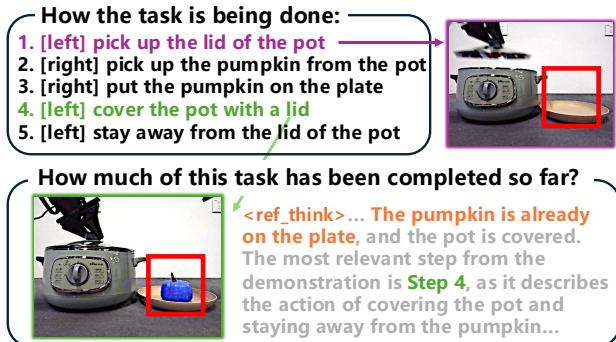


Figure 8: Illustration of implicit state accumulation required by text-based demonstrations. Although Step 1 (purple) and Step 4 (green) appear action-wise similar that interact with the pot lid, they differ in whether the pumpkin has already been placed on the plate (red box). Correctly identifying the current progress stage therefore requires integrating intervening steps to recover the accumulated state.

Why are text-based demonstrations harder than vision-based ones? *Text-based demonstrations require implicit state accumulation rather than direct state matching.* Unlike vision-based demonstrations, where each step explicitly depicts a complete world state, text-based demonstrations describe actions whose effects must be cumulatively integrated to infer the underlying state. As shown in Figure 8, Step 1 and Step 4 both involve interacting with the pot lid, but they are distinguished by an implicit state variable: whether the pumpkin has already been removed from the pot and placed on the plate. Only by integrating the intervening Steps 2 and 3 can the model infer that the current observation aligns with Step 4 (green) rather than Step 1 (purple). This demonstrates that progress estimation under text-based demonstrations hinges on maintaining and updating an implicit world state over time, rather than relying on surface-level action semantics or keywords matching.

6 Related Work

Progress Estimation. Prior work on progress estimation has largely focused on task-specific or expert models trained to predict progress within fixed tasks or environments (Yang et al., 2024; Chen et al., 2025; Ma et al., 2024a), limiting their generalization. Some approaches estimate progress via latent feature distances (Ma et al., 2022, 2023; Escontrela et al., 2023; Lee et al., 2021), but they are limited in modeling fine-grained intermediate progress. More recent methods attempt to leverage the generalization ability of Vision-Language Models (VLMs), but typically formulate progress estimation indirectly (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), where progress is defined only relative to other observations within the same trajectory and remains tightly coupled to the full sequence context. In contrast, humans can infer how far a task has progressed from a single observation by reasoning over underlying task dynamics rather than local visual similarity. This view frames progress estimation as a long-horizon, dynamic reasoning problem, motivating our study of whether VLMs can acquire progress estimation as a general reasoning capability from a single observation.

Progress Reasoning in VLMs. While recent advances in vision–language models have greatly improved static visual reasoning from single or multiple images (Zhang et al., 2025; Lee et al., 2025; Yuan et al., 2025a,b; Pi et al., 2024; Cai et al., 2025), such reasoning largely focuses on snapshot-level perception, i.e., what is immediately visible. In contrast, *dynamic reasoning* under partial observation requires models to reason over long-horizon task evolution, inferring latent state transitions and

maintaining an internal representation of how the world has changed over time. Progress reasoning provides a natural and challenging testbed for this capability, as it simultaneously demands long-horizon reasoning, implicit world modeling, and fine-grained understanding of intermediate task states from a single observation. However, existing VLM-based approaches typically offer only coarse judgments of task completion or rely on indirect signals, without explicitly quantifying intermediate progress (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 study, which investigates whether general-purpose VLMs can be adapted to perform progress reasoning as a structured, step-by-step inference process, rather than as a static or heuristic prediction.

7 Conclusion

We investigated progress estimation as a form of long-horizon and dynamic reasoning that goes beyond static visual understanding. To systematically study this capability, we introduced PROGRESS-BENCH, which evaluates progress reasoning from a single observation under controlled variations of demonstration modality, viewpoint, and answerability. Inspired by human cognition, we further explored a two-stage progress reasoning paradigm based on episodic retrieval and mental simulation, exploring such reasoning in training-free and training-based approaches. Our experiments across 14 VLMs show that current models struggle with progress estimation that are highly sensitive to modality and viewpoint changes, often produce coarse or collapsed progress scores, and fail to properly handle unanswerable cases. We also find that training-free prompting provides only limited and model-dependent improvements. In contrast, by explicitly training models to internalize this process, PROGRESSLM-3B achieves consistent gains in accuracy, robustness, and uncertainty handling, even at small model scale. We hope our benchmark and insights facilitate future research on reasoning about the world from a long-horizon and dynamic perspective.

References

- Minttu Alakuijala, Reginald McLean, Isaac Woungang, Nariman Farsad, Samuel Kaski, Pekka Marttinen, and Kai Yuan. Video-language critic: Transferable reward functions for language-conditioned robotics. *Transactions on Machine Learning Research (TMLR)*, 2025.
- Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.
- Zhipeng Cai, Ching-Feng Yeh, Hu Xu, Zhuang Liu, Gregory Meyer, Xinjie Lei, Changsheng Zhao, Shang-Wen Li, Vikas Chandra, and Yangyang Shi. Depthlm: Metric depth from vision language models. *arXiv preprint arXiv:2509.25413*, 2025.
- Qianzhong Chen, Justin Yu, Mac Schwager, Pieter Abbeel, Fred Shentu, and Philipp Wu. Sarm: Stage-aware reward modeling for long horizon robot manipulation, 2025. URL <https://arxiv.org/abs/2509.25358>.
- Yuchen Cui, Scott Niekum, Abhinav Gupta, Vikash Kumar, and Aravind Rajeswaran. Can foundation models perform zero-shot task specification for robot manipulation? In *Learning for dynamics and control conference*, pp. 893–905. PMLR, 2022.
- Yinpei Dai, Jayjun Lee, Nima Fazeli, and Joyce Chai. Racer: Rich language-guided failure recovery policies for imitation learning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 15657–15664. IEEE, 2025.
- Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.
- Jiafei Duan, Wilbert Pumacay, Nishanth Kumar, Yi Ru Wang, Shulin Tian, Wentao Yuan, Ranjay Krishna, Dieter Fox, Ajay Mandlekar, and Yijie Guo. Aha: A vision-language-model for detecting and reasoning over failures in robotic manipulation. *arXiv preprint arXiv:2410.00371*, 2024.
- Alejandro Escontrela, Ademi Adeniji, Wilson Yan, Ajay Jain, Xue Bin Peng, Ken Goldberg, Youngwoon Lee, Danijar Hafner, and Pieter Abbeel. Video prediction models as rewards for reinforcement learning. *Advances in Neural Information Processing Systems*, 36:68760–68783, 2023.
- Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge. *Advances in Neural Information Processing Systems*, 35: 18343–18362, 2022.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.
- Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report, 2025a. URL <https://arxiv.org/abs/2508.02324>.

Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, Zhuqin Yang, Meng Li, Yinuo Zhao, Zhiyuan Xu, Guang Yang, et al. Robomind: Benchmark on multi-embodiment intelligence normative data for robot manipulation. In *Robotics: Science and Systems (RSS) 2025*. Robotics: Science and Systems Foundation, 2025b. URL <https://www.roboticsproceedings.org/rss21/p152.pdf>.

Daniel Yang, Davin Tjia, Jacob Berg, Dima Damen, Pulkit Agrawal, and Abhishek Gupta. Rank2reward: Learning shaped reward functions from passive video. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2806–2813. IEEE, 2024.

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

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

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

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

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

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

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

Appendix

A Data Construction Details	17
A.1 Text Unanswerable Data	17
A.2 Vision-Based Unanswerable Data	18
A.3 Chain-of-Thought Generation	19
A.4 Human Bench Data	20
B Experimental Settings	21
B.1 Model Inference.	21
B.2 Text-Based Unanswerable Data Generation.	21
B.3 Vision-Based Unanswerable Data Generation.	22
B.4 Chain-of-Thought Data Generation	22
B.5 Supervised Fine-Tuning.	23
B.6 Reinforcement Learning.	23
C Supplementary Results and Analysis	24
C.1 Vision-Based Demo Case Studies.	24
C.2 Text-Based Demo Case Studies.	26
C.3 Analysis of Coupled Progress Two-Stage Reasoning	27
C.4 In the Wild Generalization Analysis	28
C.5 Unanswerable Case Recognition	30
D Prompts	31
D.1 Vision-based Demo	31
D.2 Text-based Demo	31
D.3 Vision-based Chain-of-Thought Prompt	31
D.4 Text-based Chain-of-Thought Prompt	31
D.5 Unanswerable Vision-based Sample Generation	31
D.6 Unanswerable Text-Based Sample Generation	31

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 $\{s_1, s_2, \dots, s_n\}$, 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

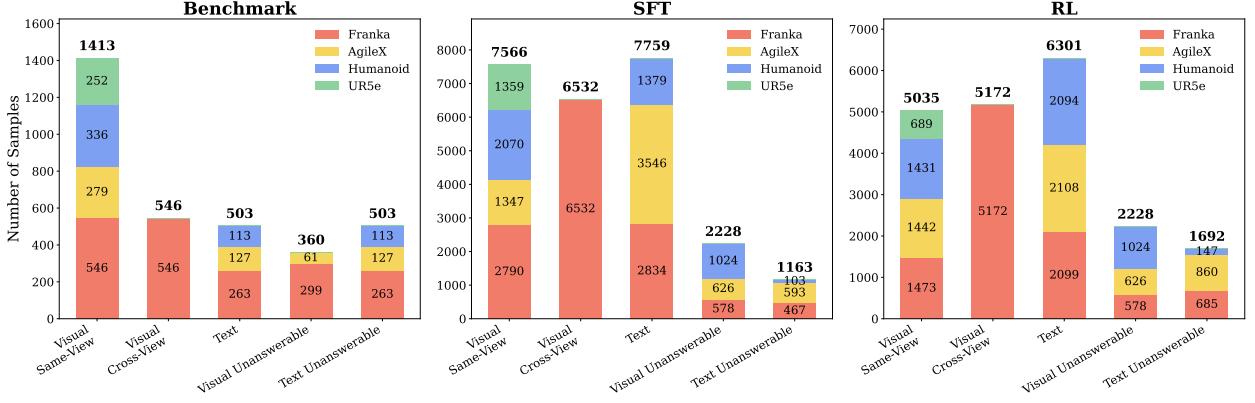


Figure 9: Data distribution statistics across Benchmark, SFT, and RL splits. This figure shows the distribution of samples produced by our data construction pipeline across Benchmark, Supervised Fine-Tuning (SFT), and Reinforcement Learning (RL) stages. Samples are organized by demonstration–observation 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.

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 $\{s_1, s_2, \dots, s_n\}$, 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 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

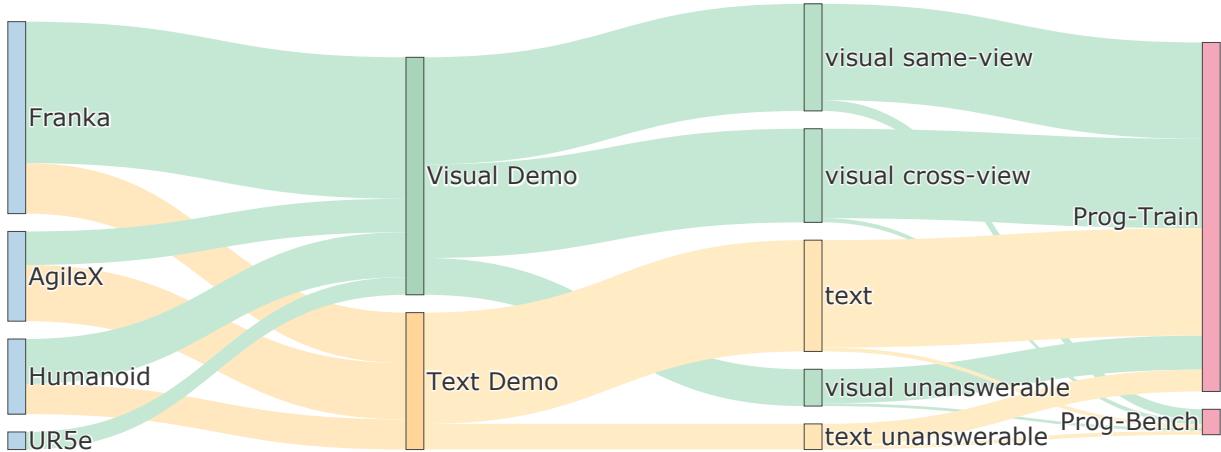


Figure 10: **Data construction pipeline.** This Sankey diagram illustrates how raw manipulation trajectories from four heterogeneous robotic platforms (Franka, AgileX, Humanoid, and UR5e) are transformed through our data construction process. Demonstrations are first organized into visual and text modalities, 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.

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

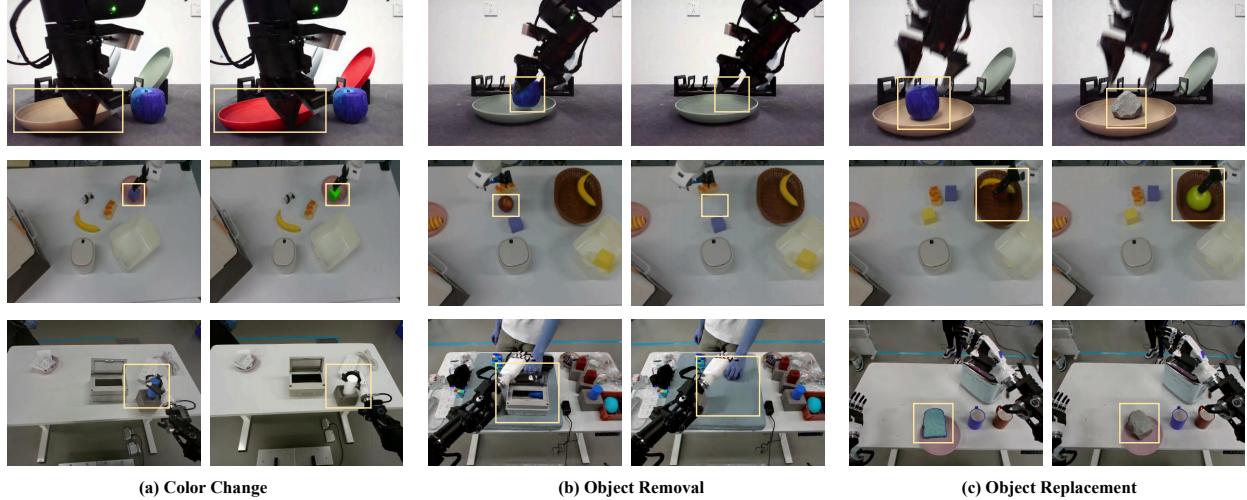


Figure 11: Case Visualization of visual unanswerable samples construction via Image Editing. To test whether models can recognize ill-defined progress, we construct visual unanswerable samples by breaking the semantic consistency between demonstrations and observations while preserving realism. Given 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.

sequence of demonstration frames with associated progress values (e.g., “<image> 0% <image> 25% <image> 50% <image> 75% <image> 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**:

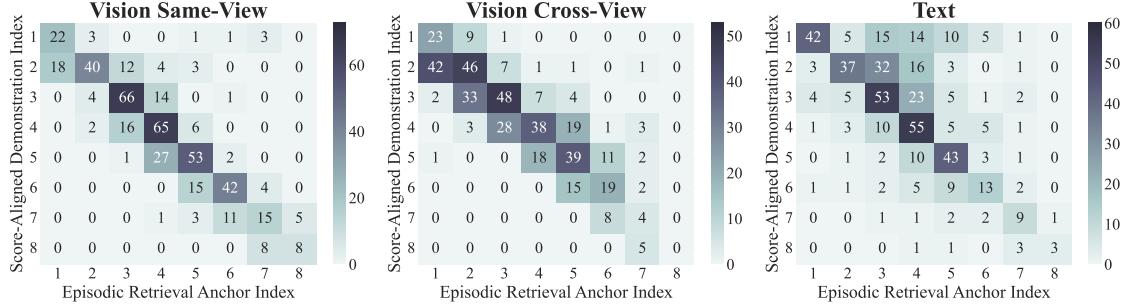


Figure 12: Diagnostic analysis of coupled progress reasoning. Heatmaps show the relationship between the episodic retrieval anchor index (x-axis) and the score-aligned demonstration index (y-axis) under Vision Same-View, Vision Cross-View, and Text settings. A strong diagonal indicates tight coupling 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.

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 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.

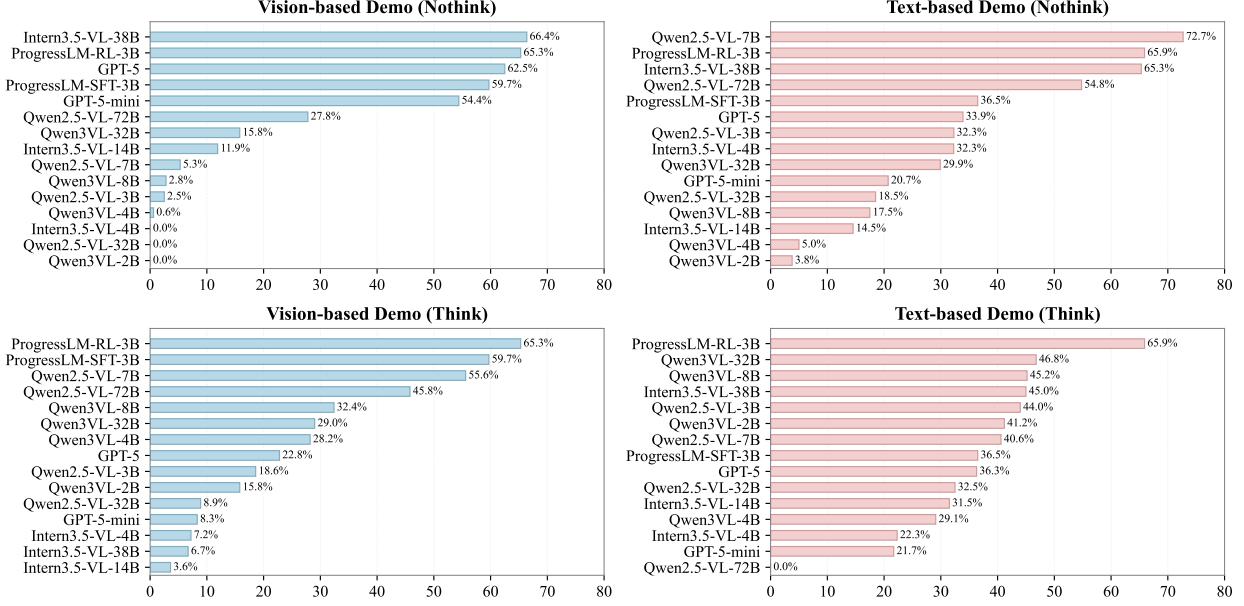


Figure 13: Unanswerable Detection Accuracy (UDA) across models with and without training-free thinking. This figure compares unanswerable detection accuracy under Vision-based and Text-based demonstrations, contrasting standard inference (NoThink) with training-free explicit reasoning (Think). Across 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.

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 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 multimodal 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.

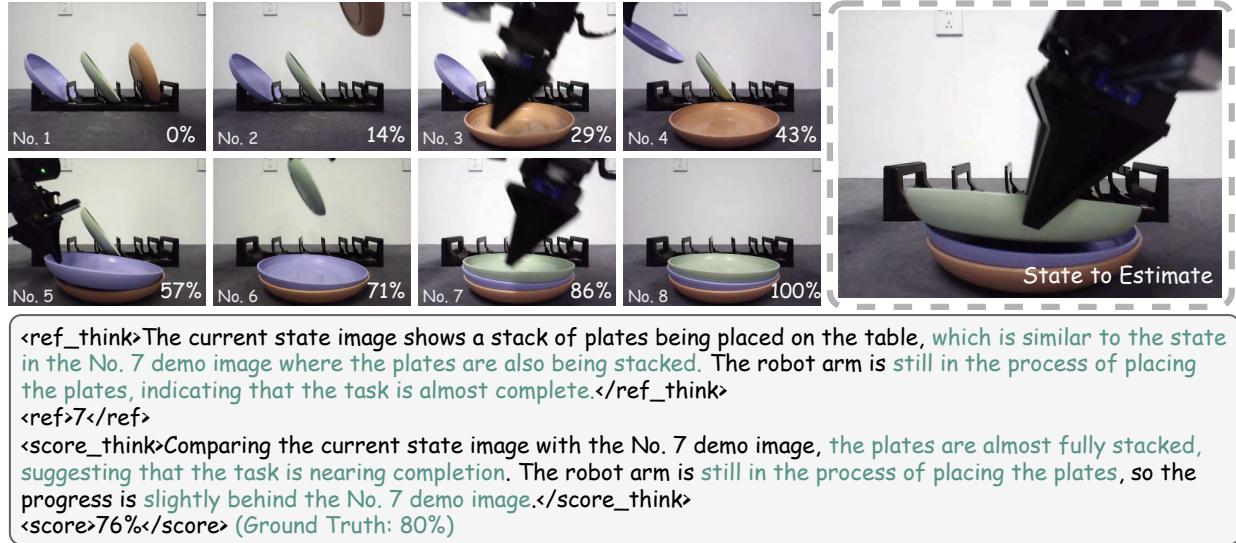


Figure 14: **Vision-Based Case Visualization (Same-View)**. This example illustrates how the model performs progress estimation by coupling episodic retrieval with mental simulation. Given a current observation (right), the model retrieves the most semantically aligned demonstration step (No. 7) from 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.

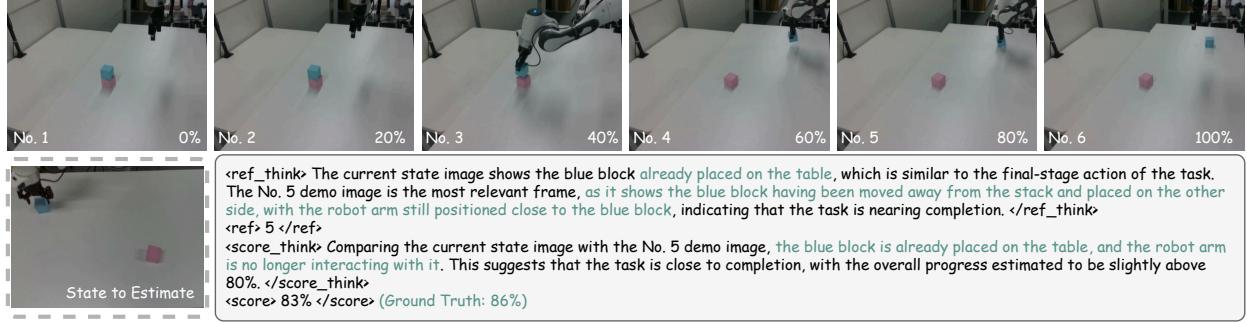


Figure 15: Vision-Based Case Visualization (Cross-View). This example illustrates progress estimation with cross-view visual demonstrations, where the observation is captured from a viewpoint different from the demonstration sequence. Given the current state image (bottom left), the model retrieves the 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.

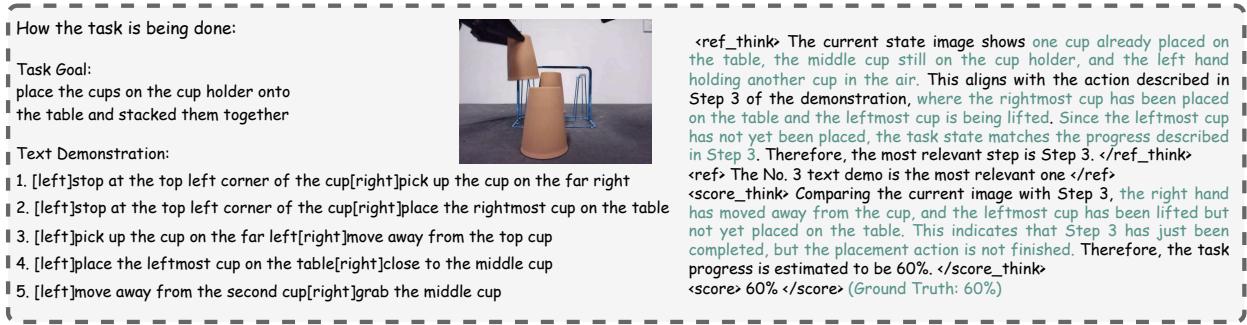


Figure 16: Text-Based Case Visualization. This example illustrates progress estimation with text-based demonstrations. Given the current visual observation, the model retrieves the most semantically aligned textual step (Step 3) from the demonstration by matching object states and action semantics described 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.

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.

Table 3: Human Bench: Comparison of in-the-wild model generalization performance on Visual and Textual settings. 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% <small>-1.8</small>	78.5% <small>+7.6</small>	0.2% <small>-0.1</small>	23.8% <small>+1.9</small>	75.3% <small>-2.9</small>	1.9% <small>+4.1</small>	23.7% <small>+0.1</small>	76.9% <small>+2.4</small>	1.1% <small>+2.0</small>
Qwen2.5VL-32B	31.7% <small>-4.3</small>	50.0% <small>+22.7</small>	0.0%	26.7% <small>+0.6</small>	70.5% <small>-1.4</small>	0.2% <small>+0.2</small>	29.2% <small>-1.8</small>	60.2% <small>+10.6</small>	0.1% <small>+0.1</small>
Qwen2.5VL-7B	29.0% <small>+11.0</small>	52.2% <small>-11.0</small>	0.6% <small>+21.7</small>	29.3% <small>+6.1</small>	58.7% <small>-3.7</small>	11.4% <small>-4.4</small>	29.2% <small>+8.5</small>	55.4% <small>-7.3</small>	6.0% <small>+8.6</small>
Qwen2.5VL-3B	27.8% <small>+7.7</small>	48.0% <small>-12.1</small>	0.0% <small>+3.3</small>	52.7% <small>-7.3</small>	17.2% <small>+11.2</small>	23.8% <small>-12.8</small>	40.3% <small>+0.2</small>	32.6% <small>-0.4</small>	11.9% <small>-4.7</small>
Qwen3VL-32B	19.7% <small>+1.6</small>	91.7% <small>-0.7</small>	0.0% <small>+0.1</small>	23.4% <small>+0.7</small>	77.9% <small>+0.5</small>	0.1% <small>+0.2</small>	21.5% <small>+1.1</small>	84.8% <small>-0.1</small>	0.04% <small>+0.1</small>
Qwen3VL-8B	19.6% <small>+4.3</small>	80.9% <small>+3.3</small>	0.0% <small>+0.7</small>	25.0% <small>+0.3</small>	69.3% <small>+3.8</small>	0.1% <small>+4.8</small>	22.3% <small>+2.3</small>	75.1% <small>+3.5</small>	0.04% <small>+2.8</small>
Qwen3VL-4B	22.4% <small>+1.6</small>	80.4% <small>+1.3</small>	0.0% <small>+0.9</small>	25.7% <small>-0.9</small>	68.5% <small>+0.9</small>	0.0% <small>+1.1</small>	24.0% <small>+0.3</small>	74.4% <small>+1.1</small>	0.0% <small>+1.0</small>
Qwen3VL-2B	48.4% <small>+2.8</small>	33.6% <small>+11.6</small>	0.1% <small>+13.2</small>	67.4% <small>-17.6</small>	5.6% <small>+32.2</small>	0.0% <small>+10.5</small>	57.9% <small>-7.4</small>	19.6% <small>+21.9</small>	0.04% <small>+11.9</small>
Intern3.5-VL-38B	40.8% <small>+17.6</small>	44.5% <small>-8.9</small>	14.2% <small>-10.3</small>	25.9% <small>-0.1</small>	59.7% <small>+2.0</small>	7.7% <small>-3.3</small>	33.3% <small>+8.7</small>	52.1% <small>-3.4</small>	10.9% <small>-6.8</small>
Intern3.5-VL-14B	34.4% <small>+24.8</small>	52.2% <small>-24.9</small>	8.0% <small>-7.8</small>	30.0% <small>-1.4</small>	53.8% <small>+3.2</small>	0.6% <small>+1.2</small>	32.2% <small>+11.7</small>	53.0% <small>-10.9</small>	4.3% <small>-3.3</small>
Intern3.5-VL-4B	34.7% <small>+15.2</small>	50.3% <small>-20.1</small>	0.0% <small>+0.1</small>	33.6% <small>-4.5</small>	35.3% <small>+16.7</small>	0.9% <small>+0.6</small>	34.2% <small>+5.4</small>	42.8% <small>-1.7</small>	0.5% <small>+0.4</small>
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%

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 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.

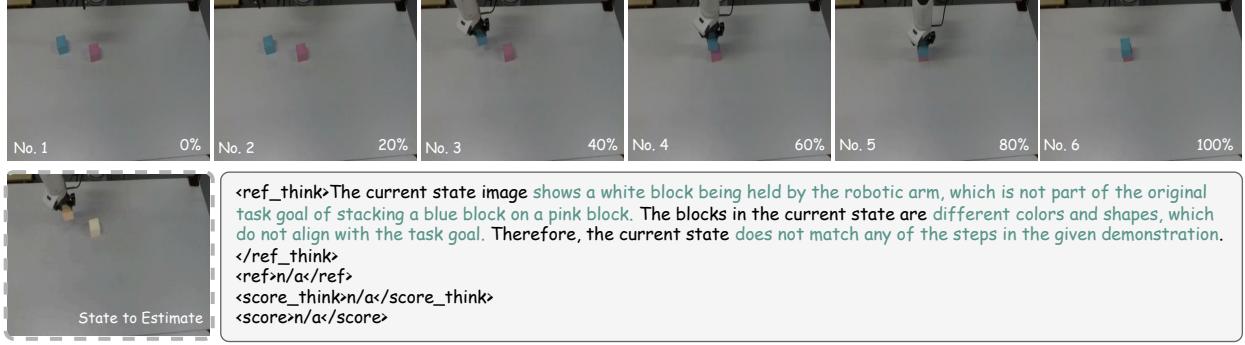


Figure 17: **Visual Unanswerable Case Visualization.** This example illustrates a visual unanswerable scenario where the current observation is semantically inconsistent with the given demonstration. While the demonstration depicts a task of stacking a blue block on a pink block, the observed state shows 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.

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 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.

How the task is being done:

Task Goal:
place the bowls on the bowl holder onto the table, and stacked together

Text Demonstration:

1. [right]pick up the bowl on the far right
2. [right]place the rightmost bowl on the table
3. [left]pick up the bowl on the far left
4. [left]place the leftmost bowl on the first bowl
5. [left]pick up the bowl on the far left
6. [left]place the leftmost bowl on the second bowl

<ref_think> The current image shows a stack of cups on the floor, which is entirely different from the task goal of stacking plates on a plate holder. The objects in the image (cups) do not align with the objects mentioned in the task goal (plates). Additionally, the image does not correspond to any step in the demonstration, which involves plates and a plate holder. Therefore, the current state image cannot be matched to any step in the demonstration. </ref_think>

<ref>n/a</ref>

<score_think>n/a</score_think>

<score>n/a</score>

Figure 18: 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.

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 estimation, 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 correspond 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.

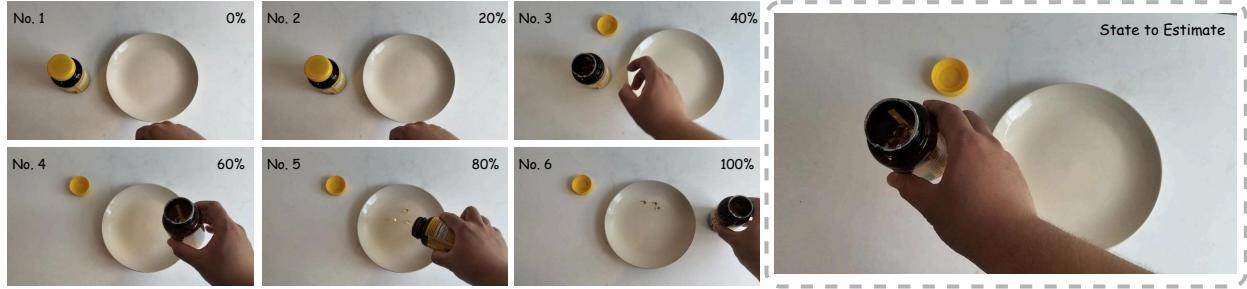
C.4 In the Wild Generalization Analysis

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 conditions, 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



<ref_think> The current state image shows the jar being opened, which is similar to the action in the No. 3 demo image where the jar is also opened. The hand is holding the jar and has lifted it slightly off the table, while the lid is removed, indicating that the task is progressing toward the next step of pouring the contents into the bowl. </ref_think>
 <ref> 3 </ref>
 <score>43%</score> (Ground Truth: 41%)
 <score>43%</score> Comparing the current state image with the No. 3 demo image, the jar is already opened, and the hand is holding it with the jar lifted away from the table, which suggests that the task is slightly more advanced than the No. 3 demo image. However, the contents have not yet been poured into the bowl, and the jar remains positioned on the left side of the plate, so the progress is estimated to be slightly more than 40% but not reaching 60%. </score_think>

Figure 19: 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.

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 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 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.

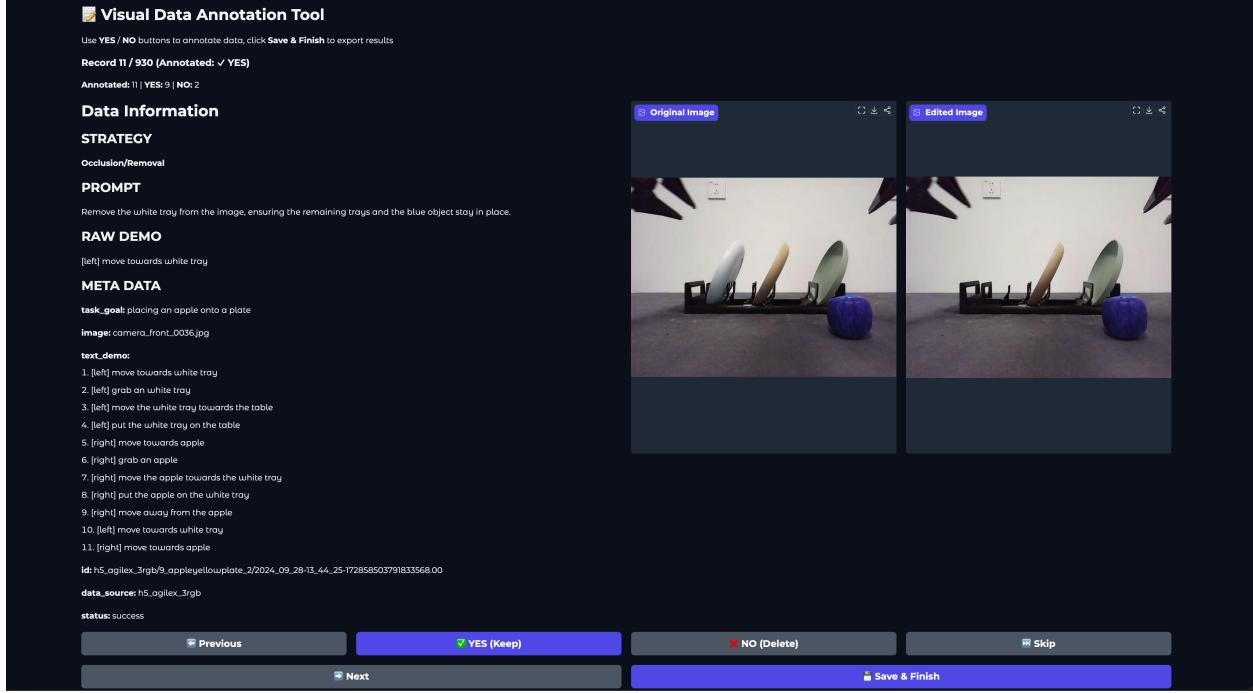


Figure 20: Gradio-based Human Filtering Platform for Visual Unanswerable Data Generation. We employ a Gradio-based annotation interface to manually verify the quality of edited images used for visual unanswerable construction. Annotators are presented with the original and edited images alongside the 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.

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 mentioned 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 encountering 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

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