PROTOWORLD: A BENCHMARK FOR WORLD MODELING FROM LABORATORY PROTOCOLS

Anonymous authors
Paper under double-blind review

ABSTRACT

Laboratory protocols specify how to perform and reproduce experiments. Developing and executing them takes expertise and time, with some experimental workflows spanning months or even years. If AI agents could automate these procedures, they could save researchers time and effort. To do so, they would first need to model how experimental actions change the contents of laboratory containers, such as tubes. Thus, we introduce ProtoWorld, a benchmark for evaluating this aspect of world modeling from laboratory protocols. We give a model the instructions from the start of a protocol through a selected step and ask either how much liquid a container holds or which experimental substances (reagents) it contains after that step. If the volume cannot be determined from the instructions, the model must answer `indeterminate`. We report a ProtoWorld Full set of 100 manually annotated questions at 82 selected steps across 43 protocol records, including a ProtoWorld Hard subset of 40 questions manually labeled as hard. We use a large language model (LLM) to extract operations such as additions, transfers, and removals from the protocol. Then, we manually review these operations, and apply a deterministic program to these operations to calculate the reference answers. Across thirteen model configurations, the highest overall accuracy is 64.7% on ProtoWorld Full and 30.0% on ProtoWorld Hard. We hope ProtoWorld will help researchers evaluate and improve the world-modeling capabilities that agents need to design, refine, and execute laboratory protocols, ultimately helping accelerate scientific discovery.

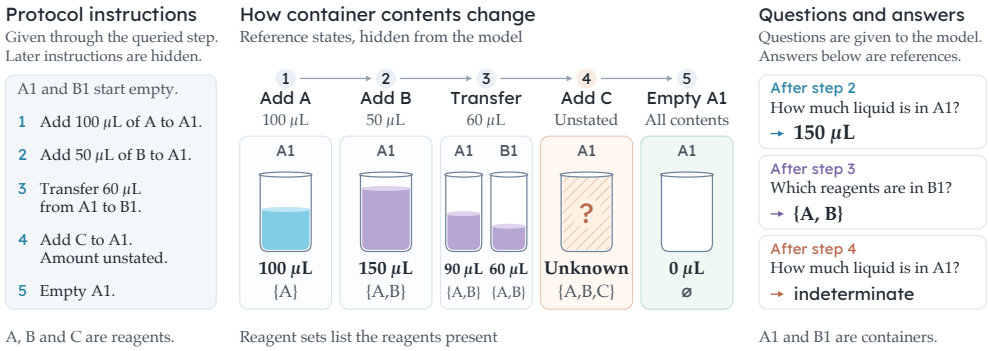


Figure 1: **World modeling from laboratory instructions.** For each question, the model receives the protocol's title and introductory text and instructions only through the queried step. The middle panel shows reference states, hidden from the model; the right pairs questions with reference answers. Volumes are in microliters (μL). Transfers move both liquid and reagents, an addition with an unstated amount makes volume indeterminate, and emptying restores a known state.

1 INTRODUCTION

Laboratory *protocols* list the materials and steps needed to perform and repeat an experiment. Developing and carrying them out takes expertise and time. Some experimental workflows can take

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

months or even years (Rapp et al., 2024). Thus, researchers are exploring how language models can follow instructions to control lab equipment (Yoshikawa et al., 2023; Pagel et al., 2026; Liu et al., 2026; Zou et al., 2026). If AI agents could design, refine, and execute protocols, they could save researchers a lot of time and effort. As a first step, an agent needs a *world model* of the experiment. This model should represent the materials in the experiment and predict how each step changes them.

Consider a protocol that adds a reagent (a substance used in an experiment) to a *well*, a small compartment for holding liquid in a laboratory plate called a *microplate*. The protocol then washes the well and transfers liquid to another well. To determine what is transferred to the second well, the model must track what remains in the first well after the addition and washing steps. The model may need to connect instructions that are many steps apart, even when some quantities are not specified. For example, "add reagent A" does not make it clear how much to add, so the well's volume cannot be determined. If a later step empties the well, no liquid remains, and thus a world model should recognize that the volume is known again and track subsequent changes from a known-empty state. Alternatively, transferring a stated amount into an empty second well makes that well's volume known, even if the first well's volume remains unknown. We describe a container's state by its liquid volume and *reagent set*, the reagents it contains.

We introduce **ProtoWorld**, a benchmark for evaluating *world modeling* by testing whether models can infer how experimental actions change container contents.[1] The model receives the protocol's title and introductory text, followed by the instructions from the first step through a selected step, called the *query point*. We then ask for the total liquid volume or reagent set in a specified container, such as a tube or microplate well, after that step. If the volume cannot be determined from the instructions, the model must answer `indeterminate`.

To demonstrate how the task works, we show an example in Figure 1. Containers A1 and B1 are initially empty. Adding 100 μL of reagent A and 50 μL of reagent B to A1 gives a total volume of 150 μL. Next, the protocol transfers 60 μL from A1 to B1. The transferred liquid contains both reagents, so B1's reagent set is $\{A, B\}$. The protocol then adds reagent C to A1 without specifying the amount, leaving A1's volume indeterminate. When the final step empties A1, no liquid remains and its volume is known to be zero. For each question shown on the right, the model receives the protocol's title and introductory text, along with the instructions on the left through the queried step. It must infer the answer without seeing the container states in the middle or the reference answers on the right. Thus, these questions test whether a model can track changes in container contents and recognize when a volume cannot be determined.

We report two nested collections. **ProtoWorld Full** contains 100 manually annotated questions, comprising 50 determinate-volume, 34 reagent-set, and 16 indeterminate-volume questions. **ProtoWorld Hard** contains the 40 questions in Full that we manually label as hard, with 12, 20, and 8 questions in these categories, respectively. ProtoWorld Full spans 43 protocol records and 82 query points across nine categories of laboratory procedures, including sequencing, immunoassays, and protein purification. All 40 questions in ProtoWorld Hard are included in ProtoWorld Full, with identical protocol contexts and reference answers.

Constructing the references requires interpreting the source instructions and keeping track of their consequences across containers. We first use an LLM to extract *structured operations* from the protocol. Each operation specifies the action, containers, reagents, and stated quantities. After manually reviewing and revising these operations, we use a deterministic program called the *ledger* to apply them in order and calculate each container's volume and reagent set. We then review the resulting reference answers and make any necessary corrections. Finally, we score model predictions against these answers using fixed rules.

We evaluate thirteen model configurations, where GPT-6 Astra achieves the highest accuracy on the 84 answerable questions (determinate volume or reagent set) in ProtoWorld Full at 71.8%, while Claude Opus 5 achieves the highest accuracy on the 32 answerable questions in ProtoWorld Hard at 37.5%. Their overall accuracies, which also include indeterminate-volume questions, are 64.7% and 30.0%, respectively. We report accuracy for each question type. We then examine when models answer `indeterminate` and whether they answer related questions correctly across repeated runs.

[1] `https://anonymous.4open.science/w/protoworld-explorer-49D1/`

Our contributions are:

- **Benchmark for evaluating world modeling.** We introduce ProtoWorld to test whether models can track container contents through experimental steps and recognize when a state cannot be determined. It includes 100 manually annotated questions in Full and a 40-question Hard subset.
- **Evaluation of thirteen models.** We compare their performance on ProtoWorld Full and ProtoWorld Hard across volume, reagent-set, and indeterminate-volume questions.
- **Analysis of model failures.** We find that models can achieve high volume accuracy while often predicting the wrong reagent set. In a case study, models also recognize an indeterminate volume but later give a number even though the volume still cannot be determined.

2 THE PROTOWORLD BENCHMARK

ProtoWorld evaluates world modeling through questions about how experimental actions change container contents. Given a protocol, a model must infer a container's volume or reagent set and recognize when the instructions do not determine the volume.

2.1 TASK DEFINITION

To evaluate this capability, we give a model the protocol's title and introductory text, followed by the instructions from the first step through query point k. We call this text the *protocol prefix*, denoted $P_{0:k}$. We then specify a target container w, such as a well or tube, and a query type $q \in \{\text{volume}, \text{reagents}\}$. The model must report the container's contents after step k. The requested output is

$$f(P_{0:k}, w, q) = \begin{cases} V_w(k), & q = \text{volume, volume is determinate,} \\ \texttt{indeterminate}, & q = \text{volume, volume is indeterminate,} \\ C_w(k), & q = \text{reagents,} \end{cases}$$

where $V_w(k)$ is the total liquid volume in container w after step k, and $C_w(k)$ is its reagent set. This set records which reagents are present, without their concentrations. For reporting, we separate volume queries into determinate-volume and indeterminate-volume questions. Tables label the determinate-volume category as Volume.

Every request includes the same system prompt, which gives rules for interpreting operations and deciding when to answer `indeterminate`. Under these rules, washing empties a well, and a transfer of a stated amount adds that amount to the destination even if the source's total volume is unknown. The model sees neither later protocol steps nor the structured operations and reference answers used for scoring.

A question is answerable when the visible instructions and these rules uniquely determine its answer. For a volume question, a missing quantity matters only if it still affects the target container at the query point. For example, adding an unspecified amount makes the volume unknown, but emptying the container later makes it known again. Models must track these changes to decide whether to give a number or answer `indeterminate`.

2.2 BENCHMARK CONSTRUCTION

Source protocols. ProtoWorld is constructed from publicly available laboratory protocols hosted on protocols.io.[2] Source protocols and construction records are available through our anonymous benchmark explorer.

Manual annotation and review. We first use an LLM to extract structured operations from the protocol text. Each record specifies an action, its containers, reagents, and stated quantities. The authors then manually review and revise these records and annotate all 100 questions in ProtoWorld Full.

[2] `https://www.protocols.io/`

Table 1: **ProtoWorld collections.**

Collection	Questions	Protocols	Query points
ProtoWorld Full	100	43	82
ProtoWorld Hard	40	21	33

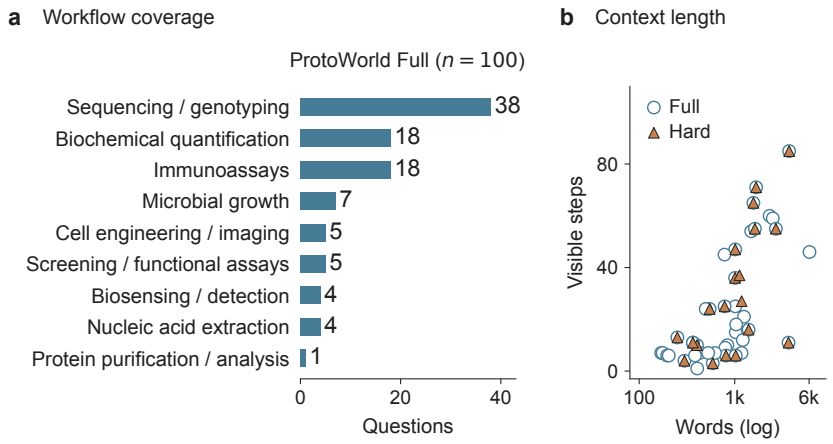


Figure 2: **ProtoWorld collections.** **a**, Distribution of the 100 ProtoWorld Full questions across nine categories of laboratory procedures. **b**, Context length for the longest included prefix of each protocol, computed separately for ProtoWorld Full and ProtoWorld Hard. Context includes the protocol's title, introductory text, and visible steps. Circles represent the 43 protocol records in Full, and triangles represent the 21 records in Hard. The horizontal axis uses a logarithmic scale.

Computing reference states. Next, we use a deterministic program, the *ledger*, to apply the reviewed operations in order through the query point. It updates each container's liquid volume and reagent set after additions, transfers, and removals. It also tracks when an unspecified quantity makes a volume unknown and when complete emptying makes that volume known again. We use only information available at or before the query point. Finally, we apply any answer corrections recorded during manual review to obtain the reference answers. Full and Hard use the same rules for interpreting operations and scoring answers. To illustrate how these steps produce the reference answers, Figure 4 in Appendix C shows the construction process.

2.3 Dataset Overview

Protocol coverage. ProtoWorld Full contains 100 questions at 82 query points from 43 protocol records, each corresponding to a source entry. The protocols span nine categories of laboratory procedures, including sequencing, immunoassays, and protein purification. To summarize this coverage, Figure 2a shows how the questions are distributed across categories.

Questions and nested collections. ProtoWorld Full contains 50 volume, 34 reagent-set, and 16 indeterminate-volume questions. The 40 questions in ProtoWorld Hard include 12 volume, 20 reagent-set, and 8 indeterminate-volume questions from 21 protocol records and 33 query points. Every Hard question also appears in Full with the same protocol context and reference answer. Table 1 summarizes the sizes of the two collections.

Context length. To describe how much text models must process, we measure the longest protocol prefix included for each protocol in each collection. These lengths include the title, introductory text, and steps visible to the model. Figure 2b plots the word and step counts. Full has a median of 811 words and 12 steps, with maxima of 6,060 words and 85 steps. Hard has a median of 1,012 words and 24 steps, with maxima of 3,712 words and 85 steps.

2.4 SCORING

We compare each prediction with its reference answer using fixed rules. Numerical volume answers must fall within a fixed tolerance of the reference volume. Reagent sets must match exactly after standardizing reagent names and applying the released alias map, which lists names treated as equivalent. Indeterminate-volume questions require the answer `indeterminate`. Missing answers, responses that cannot be parsed, and answers of the wrong type are scored as incorrect. Appendix C gives the numerical tolerance and name-normalization rules.

Overall accuracy is the fraction of questions answered correctly across all three question types. Answerable accuracy covers only determinate-volume and reagent-set questions, totaling 84 in Full and 32 in Hard. We also report accuracy for each question type. For indeterminate-volume questions, we call this accuracy *indeterminate recall*, the fraction correctly answered with `indeterminate`.

3 EVALUATION

We compare model performance on ProtoWorld Full and ProtoWorld Hard using overall accuracy and accuracy for each question type.

3.1 EXPERIMENTAL SETUP

We evaluate ProtoWorld Full and ProtoWorld Hard using the task interface and scoring rules in Section 2. All model configurations are evaluated on the same questions within each collection, comprising 100 questions at 82 query points in Full and 40 questions at 33 query points in Hard. Models receive the protocol's title and introductory text, the steps up to the query point, and the target container and question. Questions at the same query point are answered together in a single request. Future steps and reference representations are hidden.

Models and inference settings. The main comparison uses three runs for each of thirteen model configurations (model and inference mode): GPT-5.5 at `high` reasoning effort, Kimi-K2.5 in its default thinking mode, DeepSeek-V4-Flash with explicit `high` effort, Claude Sonnet 5 and Claude Opus 5 in their default adaptive-thinking modes, Gemini 3.8 Flash, Kimi-K3 and GLM-5.3 in their default thinking modes, Nemotron-3-Ultra (using the NVFP4 reduced-precision format) in its default reasoning mode, GPT-6 Astra with `high` reasoning effort, Qwen3.8-Max (0902 snapshot) in its default thinking mode, Llama-3.3-70B-Instruct (Llama-3.3-70B below), and Grok-4.1-fast (the `grok-4-1-fast-non-reasoning` deployment). The latter two use their instruction and non-reasoning interfaces without an effort setting.

Sampling and response collection. Requests specify a temperature of 1.0, top-p of 1.0, and a maximum output length of 32,000 tokens. Providers may override sampling parameters in reasoning mode, and the same output limit does not imply equal inference compute. Appendix A provides configuration-specific inference and request-delivery details. All questions remain in the evaluation denominator, and missing or invalid answers are scored as incorrect. Tables report the mean and sample standard deviation across the three runs, and figures use the same runs for the configurations shown.

3.2 FULL AND HARD RESULTS

Tables 2 and 3 report results on ProtoWorld Full and ProtoWorld Hard, respectively. Answerable accuracy measures reconstruction accuracy over determinate volume and reagent-set questions. Overall accuracy additionally includes indeterminate-volume questions. We report the three question types separately to distinguish state reconstruction from recognition of insufficient information.

On ProtoWorld Full, GPT-6 Astra achieves the highest mean answerable accuracy at 71.8%, followed by GPT-5.5 at 71.4%. Both configurations achieve the highest overall accuracy of 64.7%. High volume accuracy does not imply that a model also identifies which reagents remain. Claude Opus 5 has the highest volume accuracy at 83.3%, but scores 39.2% on reagent sets; even the highest reagent-set accuracy, achieved by GPT-5.5, is 60.8%. Reagent-set scoring requires every member

Table 2: **Results on ProtoWorld Full.** The set contains 50 volume, 34 reagent-set, and 16 indeterminate-volume questions. Answerable accuracy weights the 84 answerable questions equally, and Overall covers all 100 questions. Values are percentage means ± sample standard deviations across three runs. Rows are ordered by answerable accuracy. In each accuracy column, the highest mean is bold and the second-highest is underlined, including ties. Cost averages the requests covering the evaluated query points across three runs, using recorded token usage and fixed accounting rates (Appendix A).

Model (mode)	Answerable accuracy	Volume	Reagent set	Indeterminate recall	Overall	Cost ($/run)
GPT-6 Astra (high)	**71.8 ± 1.8**	80.7 ± 1.2	58.8 ± 2.9	27.1 ± 3.6	**64.7 ± 1.5**	3.91
GPT-5.5 (high)	71.4 ± 3.1	78.7 ± 2.3	**60.8 ± 4.5**	29.2 ± 3.6	**64.7 ± 2.3**	9.86
Qwen3.8-Max (thinking)	68.7 ± 4.8	78.0 ± 5.3	54.9 ± 6.1	20.8 ± 3.6	61.0 ± 4.0	6.17
Gemini 3.8 Flash (thinking)	66.7 ± 1.2	75.3 ± 2.3	53.9 ± 1.7	37.5 ± 0.0	62.0 ± 1.0	1.34
Claude Opus 5 (adaptive)	65.5 ± 4.3	**83.3 ± 4.2**	39.2 ± 4.5	6.2 ± 6.2	56.0 ± 4.6	2.69
Claude Sonnet 5 (adaptive)	61.1 ± 4.8	76.7 ± 1.2	38.2 ± 12.8	10.4 ± 3.6	53.0 ± 4.4	2.42
Kimi-K2.5 (thinking)	61.1 ± 4.8	76.7 ± 2.3	38.2 ± 8.8	22.9 ± 3.6	55.0 ± 4.0	3.54
Kimi-K3 (thinking)	58.3 ± 2.1	81.3 ± 1.2	24.5 ± 4.5	18.8 ± 0.0	52.0 ± 1.7	5.50
GLM-5.3 (thinking)	48.0 ± 3.0	74.7 ± 4.6	8.8 ± 2.9	12.5 ± 6.2	42.3 ± 3.2	4.77
DeepSeek-V4-Flash (high)	43.7 ± 4.2	57.3 ± 4.2	23.5 ± 10.2	43.8 ± 0.0	43.7 ± 3.5	0.96
Nemotron-3-Ultra (NVFP4)	42.1 ± 10.4	47.3 ± 7.6	34.3 ± 14.8	54.2 ± 15.7	44.0 ± 10.4	1.40
Grok-4.1-fast (non-reasoning)	38.1 ± 1.2	32.0 ± 2.0	47.1 ± 2.9	8.3 ± 3.6	33.3 ± 0.6	0.11
Llama-3.3-70B (Instruct)	25.4 ± 4.2	20.0 ± 2.0	33.3 ± 9.5	**62.5 ± 0.0**	31.3 ± 3.5	0.24

Table 3: **Results on ProtoWorld Hard.** The set contains 12 volume, 20 reagent-set, and 8 indeterminate-volume questions. Answerable accuracy weights the 32 answerable questions equally, and Overall covers all 40 questions. Values are percentage means ± sample standard deviations across three runs. Rows follow the same order as Table 2. In each accuracy column, the highest mean is bold and the second-highest is underlined, including ties. Cost uses the same accounting method as Table 2, restricted to the 33 query points in ProtoWorld Hard.

Model (mode)	Answerable accuracy	Volume	Reagent set	Indeterminate recall	Overall	Cost ($/run)
GPT-6 Astra (high)	36.5 ± 1.8	41.7 ± 0.0	33.3 ± 2.9	0.0 ± 0.0	29.2 ± 1.4	1.76
GPT-5.5 (high)	36.5 ± 4.8	36.1 ± 12.7	**36.7 ± 7.6**	0.0 ± 0.0	29.2 ± 3.8	5.01
Qwen3.8-Max (thinking)	34.4 ± 5.4	38.9 ± 4.8	31.7 ± 5.8	0.0 ± 0.0	27.5 ± 4.3	3.06
Gemini 3.8 Flash (thinking)	26.0 ± 4.8	33.3 ± 8.3	21.7 ± 2.9	12.5 ± 0.0	23.3 ± 3.8	0.75
Claude Opus 5 (adaptive)	**37.5 ± 5.4**	44.4 ± 4.8	33.3 ± 5.8	0.0 ± 0.0	**30.0 ± 4.3**	1.25
Claude Sonnet 5 (adaptive)	35.4 ± 4.8	**47.2 ± 4.8**	28.3 ± 10.4	4.2 ± 7.2	29.2 ± 5.2	1.20
Kimi-K2.5 (thinking)	30.2 ± 7.9	41.7 ± 8.3	23.3 ± 10.4	0.0 ± 0.0	24.2 ± 6.3	1.58
Kimi-K3 (thinking)	27.1 ± 4.8	36.1 ± 4.8	21.7 ± 5.8	0.0 ± 0.0	21.7 ± 3.8	2.66
GLM-5.3 (thinking)	15.6 ± 5.4	22.2 ± 9.6	11.7 ± 2.9	0.0 ± 0.0	12.5 ± 4.3	3.04
DeepSeek-V4-Flash (high)	25.0 ± 5.4	36.1 ± 12.7	18.3 ± 2.9	16.7 ± 7.2	23.3 ± 5.8	0.43
Nemotron-3-Ultra (NVFP4)	13.5 ± 4.8	19.4 ± 4.8	10.0 ± 5.0	29.2 ± 26.0	16.7 ± 7.6	0.64
Grok-4.1-fast (non-reasoning)	34.4 ± 3.1	33.3 ± 0.0	35.0 ± 5.0	4.2 ± 7.2	28.3 ± 1.4	0.04
Llama-3.3-70B (Instruct)	21.9 ± 5.4	16.7 ± 0.0	25.0 ± 8.7	**62.5 ± 0.0**	**30.0 ± 4.3**	0.10

to match after normalization, so this gap does not isolate errors in tracking mixtures from naming mismatches.

On ProtoWorld Hard, Claude Opus 5 achieves the highest mean answerable accuracy at 37.5%, followed by GPT-6 Astra and GPT-5.5 at 36.5%. Claude Opus 5 and Llama-3.3-70B tie for the highest overall accuracy at 30.0%, despite different performance profiles. Their answerable accuracies are 37.5% and 21.9%, respectively, while their indeterminate recalls are 0.0% and 62.5%. The tied overall scores thus reflect different strengths: Claude Opus 5 answers more of the answerable questions correctly, while Llama-3.3-70B recognizes more indeterminate volumes.

All thirteen configurations obtain lower answerable and overall accuracies on ProtoWorld Hard than on ProtoWorld Full. The sets differ in question-type proportions, with reagent-set questions accounting for 50% of ProtoWorld Hard and 34% of ProtoWorld Full. The larger share of reagent-set questions does not fully explain the lower overall scores: all thirteen configurations also score lower

Table 4: **Answering behavior on ProtoWorld Full and ProtoWorld Hard.** False abstention is measured over 84 answerable questions in ProtoWorld Full and 32 in ProtoWorld Hard. Indeterminate recall and numerical commitment are measured over 16 and eight indeterminate-volume questions, respectively. Values are percentage means ± sample standard deviations across three runs. Rows follow the order in Table 2.

	ProtoWorld Full			ProtoWorld Hard		
Model (mode)	False abstention ↓	Indeterminate recall ↑	Numerical commitment ↓	False abstention ↓	Indeterminate recall ↑	Numerical commitment ↓
GPT-6 Astra (high)	7.1 ± 1.2	27.1 ± 3.6	72.9 ± 3.6	13.5 ± 3.6	0.0 ± 0.0	100.0 ± 0.0
GPT-5.5 (high)	11.9 ± 1.2	29.2 ± 3.6	70.8 ± 3.6	24.0 ± 6.5	0.0 ± 0.0	100.0 ± 0.0
Qwen3.8-Max (thinking)	13.1 ± 3.1	20.8 ± 3.6	79.2 ± 3.6	25.0 ± 3.1	0.0 ± 0.0	100.0 ± 0.0
Gemini 3.8 Flash (thinking)	14.7 ± 3.0	37.5 ± 0.0	62.5 ± 0.0	31.2 ± 3.1	12.5 ± 0.0	87.5 ± 0.0
Claude Opus 5 (adaptive)	0.4 ± 0.7	6.2 ± 6.2	93.8 ± 6.2	1.0 ± 1.8	0.0 ± 0.0	100.0 ± 0.0
Claude Sonnet 5 (adaptive)	2.0 ± 0.7	10.4 ± 3.6	89.6 ± 3.6	5.2 ± 1.8	4.2 ± 7.2	95.8 ± 7.2
Kimi-K2.5 (thinking)	7.9 ± 2.7	22.9 ± 3.6	77.1 ± 3.6	20.8 ± 7.2	0.0 ± 0.0	100.0 ± 0.0
Kimi-K3 (thinking)	2.8 ± 0.7	18.8 ± 0.0	81.2 ± 0.0	7.3 ± 1.8	0.0 ± 0.0	100.0 ± 0.0
GLM-5.3 (thinking)	2.8 ± 2.7	12.5 ± 6.2	83.3 ± 9.5	7.3 ± 7.2	0.0 ± 0.0	100.0 ± 0.0
DeepSeek-V4-Flash (high)	22.2 ± 0.7	43.8 ± 0.0	56.2 ± 0.0	33.3 ± 1.8	16.7 ± 7.2	83.3 ± 7.2
Nemotron-3-Ultra (NVFP4)	31.3 ± 3.0	54.2 ± 15.7	45.8 ± 15.7	42.7 ± 7.2	29.2 ± 26.0	70.8 ± 26.0
Grok-4.1-fast (non-reasoning)	6.0 ± 1.2	8.3 ± 3.6	89.6 ± 3.6	6.2 ± 0.0	4.2 ± 7.2	95.8 ± 7.2
Llama-3.3-70B (Instruct)	26.2 ± 5.5	62.5 ± 0.0	31.2 ± 0.0	28.1 ± 8.3	62.5 ± 0.0	37.5 ± 0.0

on Hard when both collections use the Full question-type weights (Table 6). The differences range from 4.1 to 36.0 percentage points. This adjustment accounts for type proportions, not for differences in question content. Results by state-transition depth are reported in Appendix E.

4 Analysis

To understand the errors behind these scores, we examine how models handle missing information, interpret protocol instructions, and answer related questions across repeated runs.

4.1 Recognizing Indeterminate Volumes

We first examine whether models distinguish answerable questions from volumes that the protocol leaves unspecified. To separate the two kinds of error, we report *false abstention*, the fraction of answerable questions answered with `indeterminate`, and *numerical commitment*, the fraction of indeterminate-volume questions receiving a valid numerical answer. Table 4 reports both measures alongside indeterminate recall, defined in Section 2.4. Missing or invalid responses remain in the denominators but count as neither valid numerical answers nor `indeterminate` responses.

The configurations with the highest answerable accuracies still often give numbers when the available information does not support one. On ProtoWorld Full, GPT-6 Astra and GPT-5.5 return numeric answers on 72.9% and 70.8% of indeterminate-volume questions, respectively. Claude Opus 5 has the lowest false-abstention rate at 0.4% and the highest numerical-commitment rate at 93.8%. Llama-3.3-70B achieves the highest indeterminate recall at 62.5%, with a false-abstention rate of 26.2%. Nemotron-3-Ultra follows with 54.2% indeterminate recall and has the highest false-abstention rate at 31.3%.

On ProtoWorld Hard, seven configurations return numeric answers on all eight indeterminate-volume questions in every run, including GPT-6 Astra, GPT-5.5, and Claude Opus 5. Llama-3.3-70B retains an indeterminate recall of 62.5%, while every other configuration has lower recall than on ProtoWorld Full. False-abstention rates increase for all thirteen configurations, ranging from 1.0% for Claude Opus 5 to 42.7% for Nemotron-3-Ultra.

Strong performance on answerable questions thus does not ensure recognition of missing information, even at high reasoning effort. This pattern is consistent with AbstentionBench, which finds that reasoning fine-tuning can reduce abstention and that larger reasoning budgets do not reliably improve it (Kirichenko et al., 2026). One possible explanation is that models fill missing quantities with plausible assumptions instead of preserving them as unknown. Our cross-model comparison does not establish this mechanism or isolate the effect of reasoning effort; moreover, Llama's higher

indeterminate recall accompanies more false abstention. These results motivate evaluating answerability recognition alongside state reconstruction.

4.2 From Protocol Language to State Updates

Recovering container states requires resolving how many operations a protocol instruction implies. Figure 3 shows an example in which a source well supplies 36 destination wells. Claude Sonnet 5's repeated 185 μL answer is consistent with counting only three transfers, whereas GPT-6 Astra returns the correct 20 μL in all three runs. The answers suggest a scope error but do not establish its underlying cause. Appendix D examines target-container identification and volume and reagent updates.

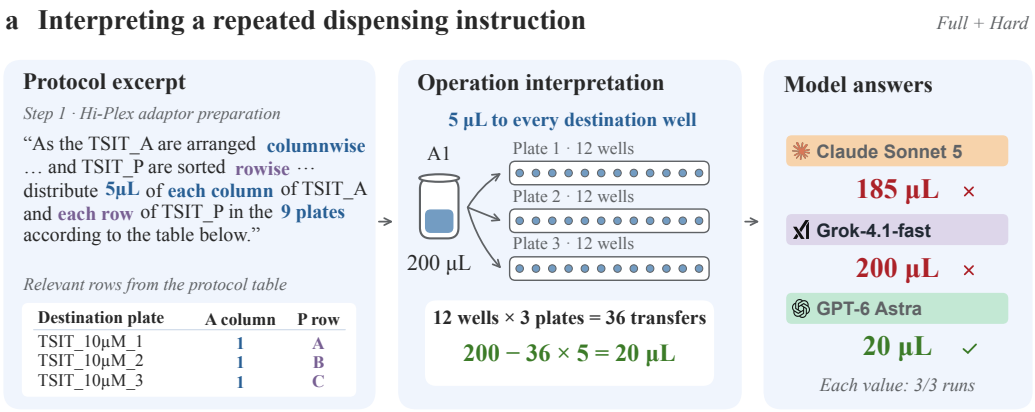


Figure 3: **Interpreting a repeated dispensing instruction.** In Hi-Plex adaptor preparation, the layout and table imply that source well A1 supplies 5 μL to 12 wells on each of three plates. Its initial 200 μL, specified in the preceding preparation table, therefore falls to 20 μL. Colored phrases connect source text to the reference interpretation. Green and red denote correct and incorrect answers across three runs. Ellipses mark omissions; three relevant rows of the nine-row table are shown.

4.3 Consistency Across Questions and Runs

We evaluate joint correctness at the 82 query points in ProtoWorld Full and the 33 query points in ProtoWorld Hard. A query point is correct in a run only if every question included at that point in the evaluated set is answered correctly. Mean joint correctness averages this rate across three runs. All-3 correct measures the proportion of query points that are jointly correct in all three runs. Both metrics weight each query point equally. Detailed results are reported in Appendix F, Figure 8.

On ProtoWorld Full, GPT-6 Astra achieves the highest mean joint correctness at 64.2% and is jointly correct in all three runs at 51 of the 82 query points (62.2%). Even for this configuration, more than a third of the query points receive an incorrect answer in at least one run. On ProtoWorld Hard, its mean joint correctness is 32.3%, and it is jointly correct in all three runs at 10 of the 33 query points (30.3%). All thirteen configurations obtain lower mean joint correctness and All-3 correct rates on ProtoWorld Hard than on ProtoWorld Full.

5 Related Work

Laboratory protocol understanding and scientific benchmarks. Wet-lab language understanding has been studied through entity and relation annotation (Kulkarni et al., 2018) and process graphs that connect actions and entities across instructions, including implicit arguments (Tamari et al., 2021). BioPlanner evaluates biological protocol planning by reconstructing pseudocode from high-level task descriptions and a supplied set of admissible functions (O'Donoghue et al., 2023); LAB-Bench and its harder successor LABBench2 evaluate a broad range of practical biology research

capabilities (Laurent et al., 2024; 2026). BioProBench grounds procedural reasoning evaluation in a large corpus of human-written biological protocols, including tasks that require quantitative precision (Liu et al., 2025). PRISM combines protocol planning with validation in a simulated laboratory and robotic execution (Hsu et al., 2026), and LabOSBench evaluates computer-use agents that operate scientific instruments through simulated interfaces (Zou et al., 2026). ProtoWorld complements these evaluations by asking what a container holds after a sequence of experimental steps, including cases where the available instructions do not determine its volume.

World models from text. Prior work models recipe actions as entity-state transformations (Bosselut et al., 2017), tests language models on tracking box contents (Kim & Schuster, 2023), and probes representations of beaker contents and entity states in text games (Li et al., 2021), as well as Othello board states (Li et al., 2022). Language models have also been evaluated as text-based world simulators that predict the next state from a given state and action, using a benchmark of text-game transitions to assess the reliability of learned simulators (Wang et al., 2024), and next-token accuracy can coexist with an incoherent world model (Vafa et al., 2024). ProtoWorld instead requires accumulating the state from protocol text and recognizing when it is undetermined.

State tracking in procedural text and in language models. ProPara tracks entity existence and locations, including unknown locations (Dalvi et al., 2018); OpenPI covers open-vocabulary state changes in instructions by predicting affected entities, attributes, and their before/after values at each step of WikiHow procedures (Tandon et al., 2020); and SCONE's Alchemy domain includes quantitative liquid updates (Long et al., 2016). Synthetic evaluations probe sequential entity states (Rezaee et al., 2025), depth-parameterized cumulative updates (Hou et al., 2026), and intermediate values across dependent tool calls (Pai & Xian, 2026). ProtoWorld tests volumes and reagent sets across laboratory transfers, partial removals, and emptying, with potentially missing quantities. Unlike LedgerAgent's model-maintained ledger for policy adherence (Uddin et al., 2026), our ledger computes reference states from reviewed operations.

Underspecification, clarification, and abstention. QuestBench formalizes missing information relative to a reasoning target and evaluates which clarification would supply sufficient information, including planning from partially observed initial states. Its tasks span logic, planning, and mathematics and can be resolved by at most one clarification (Li et al., 2026). ClarifyBench adds multi-turn clarification with a simulated user (Suri et al., 2026). AbstentionBench evaluates whether models refrain from definitive answers across diverse sources of unanswerability, including underspecification (Kirichenko et al., 2026). In ProtoWorld, missing information can cease to matter after later operations: an unspecified addition leaves a volume unknown, but emptying the container makes it known again. We test whether models recognize when later operations make a previously unknown volume determinate, as well as whether they can calculate volumes and identify remaining reagents.

6 Conclusion

We introduce ProtoWorld to evaluate world modeling from laboratory protocols through questions about container volumes, reagent sets, and indeterminate volumes. ProtoWorld Full contains 100 questions at 82 query points, including the 40-question ProtoWorld Hard subset at 33 query points. Across thirteen model configurations evaluated over three runs, the highest overall accuracies are 64.7% on ProtoWorld Full and 30.0% on ProtoWorld Hard.

Strong volume accuracy can coexist with errors in identifying remaining reagents and unsupported numerical answers. To decide what to add, remove, or transfer, an agent helping design, refine, or execute a protocol needs to know what each container holds and when its volume cannot be determined. ProtoWorld evaluates world modeling through questions about container contents and whether volumes can be determined from protocol text. A next step is to test whether higher benchmark scores translate into better decisions during protocol design and execution in a real laboratory.

AI USE STATEMENT

Generative AI tools were used in four ways. (1) Data construction: an LLM produced the initial structured operations for each protocol from its text; the authors revised these operations, and reference states were then computed deterministically from the revised operations (Section 2.2). (2) Reference review: an LLM-assisted review flagged candidate reference errors among questions that several models answered identically; removals and restorations were decided by the authors. (3) Writing and code: AI assistants were used to draft and edit text, to write analysis and figure scripts, and to run evaluation jobs; all AI-assisted text was reviewed by the authors, and all analysis code was checked against the released data. (4) AI assistants were also used to brainstorm framing and experimental directions; the authors selected, verified, and carried out the ideas that appear here. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

The benchmark is built from publicly available laboratory protocols. We release derived questions, the protocol prefixes needed to pose them, and reference states with links to the source records. The task evaluates bookkeeping of volumes and reagent identities in routine molecular-biology workflows; it involves no human subjects or personal data, and we do not foresee misuse beyond that of the underlying public protocols. Model outputs were obtained through commercial APIs under their terms of service.

REPRODUCIBILITY STATEMENT

The benchmark release will include the 100 questions in ProtoWorld Full, the 40-question ProtoWorld Hard subset, reference answers, and evaluation code (Section 2.2). The release will also include the system prompt and the requests used for evaluation. An anonymous explorer is linked in Section 1. Section 3.1 and Appendix A describe the inference settings and serving routes. Per-question outputs from all 39 main-comparison runs (13 configurations with three runs each), together with scripts for reproducing the reported tables and figures, will be released with the benchmark.

REFERENCES

Antoine Bosselut, Omer Levy, Ari Holtzman, Corin Ennis, Dieter Fox, and Yejin Choi. Simulating action dynamics with neural process networks. *arXiv preprint arXiv:1711.05313*, 2017.

Bhavana Dalvi, Lifu Huang, Niket Tandon, Wen-tau Yih, and Peter Clark. Tracking state changes in procedural text: a challenge dataset and models for process paragraph comprehension. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1595–1604, 2018.

Dengzhe Hou, Lingyu Jiang, Deng Li, Zirui Li, Fangzhou Lin, and Kazunori D Yamada. Wmf-am: Probing llm working memory via depth-parameterized cumulative state tracking. *arXiv preprint arXiv:2603.27343*, 2026.

Brian Hsu, Priyanka V Setty, Rory M Butler, Ryan Lewis, Casey Stone, Rebecca Weinberg, Thomas Brettin, Rick Stevens, Ian Foster, and Arvind Ramanathan. Prism: protocol refinement through intelligent simulation modeling. *Digital Discovery*, 5(6):2613–2628, 2026.

Najoung Kim and Sebastian Schuster. Entity tracking in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3835–3855, 2023.

Polina Kirichenko, Mark Ibrahim, Kamalika Chaudhuri, and Samuel J Bell. Abstentionbench: Reasoning llms fail on unanswerable questions. *Advances in Neural Information Processing Systems*, 38, 2026.

Chaitanya Kulkarni, Wei Xu, Alan Ritter, and Raghu Machiraju. An annotated corpus for machine reading of instructions in wet lab protocols. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 97–106, 2018.

Jon M Laurent, Joseph D Janizek, Michael Ruzo, Michaela M Hinks, Michael J Hammerling, Siddharth Narayanan, Manvitha Ponnapati, Andrew D White, and Samuel G Rodriques. Lab-bench: Measuring capabilities of language models for biology research. *arXiv preprint arXiv:2407.10362*, 2024.

Jon M Laurent, Albert Bou, Michael Pieler, Conor Igoe, Alex Andonian, Siddharth Narayanan, James Braza, Alexandros Sanchez Vassopoulos, Jacob L Steenwyk, Blake Lash, et al. Labbench2: An improved benchmark for ai systems performing biology research. *arXiv preprint arXiv:2604.09554*, 2026.

Belinda Li, Been Kim, and Zi Wang. Questbench: Can llms ask the right question to acquire information in reasoning tasks? *Advances in Neural Information Processing Systems*, 38, 2026.

Belinda Z Li, Maxwell Nye, and Jacob Andreas. Implicit representations of meaning in neural language models. In *Proceedings of the 59th annual meeting of the Association for Computational Linguistics and the 11th international joint conference on natural language processing (Volume 1: Long papers)*, pp. 1813–1827, 2021.

Kenneth Li, Aspen K Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent world representations: Exploring a sequence model trained on a synthetic task. *arXiv preprint arXiv:2210.13382*, 2022.

Yuyang Liu, Liuzhenghao Lv, Xiancheng Zhang, Li Yuan, and Yonghong Tian. Bioprobench: Comprehensive dataset and benchmark in biological protocol understanding and reasoning. *arXiv e-prints*, pp. arXiv–2505, 2025.

Zhe Liu, Jiaming Gu, Zhaohui Du, Zhe Wang, Huanbo Jin, Quan Lu, Qi Wang, Ting Xiao, Minting Pan, and Dongzhan Zhou. Protoact: Turning wet-lab protocols into embodied robotic actions. *arXiv preprint arXiv:2608.01690*, 2026.

Reginald Long, Panupong Pasupat, and Percy Liang. Simpler context-dependent logical forms via model projections. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1456–1465, 2016.

Odhran O'Donoghue, Aleksandar Shtedritski, John Ginger, Ralph Abboud, Ali Ghareeb, and Samuel Rodriques. Bioplanner: automatic evaluation of llms on protocol planning in biology. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 2676–2694, 2023.

Sebastian Pagel, Michael Jirasek, and Leroy Cronin. Verification and execution of the scientific literature via chemputation augmented by large language models. *Communications Chemistry*, 2026.

Dheeraj Mohandas Pai and Lu Xian. Long-horizon state tracking in llms: Executing md5 through a deep sequence of dependent tool calls. *arXiv preprint arXiv:2609.00012*, 2026.

Jacob T Rapp, Bennett J Bremer, and Philip A Romero. Self-driving laboratories to autonomously navigate the protein fitness landscape. *Nature chemical engineering*, 1(1):97–107, 2024.

Kiamehr Rezaee, Jose Camacho-Collados, and Mohammad Taher Pilehvar. Exploring state tracking capabilities of large language models. *arXiv preprint arXiv:2511.10457*, 2025.

Manan Suri, Puneet Mathur, Nedim Lipka, Franck Dernoncourt, Ryan A Rossi, and Dinesh Manocha. Structured uncertainty guided clarification for llm agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 40811–40838, 2026.

Ronen Tamari, Fan Bai, Alan Ritter, and Gabriel Stanovsky. Process-level representation of scientific protocols with interactive annotation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 2190–2202, 2021.

Niket Tandon, Keisuke Sakaguchi, Bhavana Dalvi, Dheeraj Rajagopal, Peter Clark, Michal Guerquin, Kyle Richardson, and Eduard Hovy. A dataset for tracking entities in open domain procedural text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6408–6417, 2020.

Md Nayem Uddin, Amir Saeidi, Eduardo Blanco, and Chitta Baral. Ledgeragent: Structured state for policy-adherent tool-calling agents. *arXiv preprint arXiv:2606.20529*, 2026.

Keyon Vafa, Justin Y Chen, Ashesh Rambachan, Jon Kleinberg, and Sendhil Mullainathan. Evaluating the world model implicit in a generative model. *Advances in Neural Information Processing Systems*, 37:26941–26975, 2024.

Ruoyao Wang, Graham Todd, Ziang Xiao, Xingdi Yuan, Marc-Alexandre Côté, Peter Clark, and Peter Jansen. Can language models serve as text-based world simulators? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1–17, 2024.

Naruki Yoshikawa, Marta Skreta, Kourosh Darvish, Sebastian Arellano-Rubach, Zhi Ji, Lasse Bjørn Kristensen, Andrew Zou Li, Yuchi Zhao, Haoping Xu, Artur Kuramshin, et al. Large language models for chemistry robotics. *Autonomous Robots*, 47(8):1057–1086, 2023.

Anqi Zou, Han Deng, Chengyu Zhang, Junquan Hu, Yu Wang, Yuxiang Xing, Aokai Zhang, Hanling Zhang, Zhaoyang Liu, Ben Fei, et al. Labosbench: Benchmarking computer use agents for scientific instrument control. *arXiv preprint arXiv:2606.16802*, 2026.

A Request Delivery and Inference Details

All requests specify temperature 1.0, top-p 1.0, and a maximum output length of 32,000 tokens. The requested seeds are 0, 1, and 2, except for Grok-4.1-fast, which uses 1, 2, and 3 because its API requires positive seeds. Parameter handling depends on the serving route, as detailed below.

Gateway routes. Claude Sonnet 5, Gemini 3.8 Flash, Kimi-K3, Nemotron-3-Ultra, GLM-5.3, and GPT-6 Astra are queried through an OpenAI-compatible gateway. The gateway discards sampling parameters and seeds for Claude Sonnet 5, which uses its default adaptive-thinking mode. It forwards temperature and top-p for the other five configurations. GPT-6 Astra uses an explicitly forwarded `high` reasoning-effort setting. GLM-5.3 responses are streamed to accommodate long generations. Nemotron-3-Ultra uses an NVFP4-quantized deployment.

Claude Opus 5 is queried through the gateway's Bedrock route in its default adaptive-thinking mode. This route discards sampling parameters and seeds and does not accept an explicit reasoning-effort field. GPT-5.5 is queried through the gateway's Azure route, which forwards temperature, top-p, and the `high` reasoning-effort setting.

OpenRouter and other configurations. Qwen3.8-Max uses the 0902 snapshot through OpenRouter, with routing restricted to Alibaba and to endpoints that declare support for all requested parameters. Responses are streamed, and no explicit reasoning-effort setting is sent. Kimi-K2.5 uses its default thinking mode. DeepSeek-V4-Flash uses explicit `high` reasoning effort. Llama-3.3-70B-Instruct and Grok-4.1-fast use their instruction and non-reasoning interfaces, respectively, without an effort setting.

Response collection and scoring. Requests that failed or returned unparseable answers were retried. All questions remain in the scoring denominators, and missing or invalid answers are counted as incorrect. Reported means and sample standard deviations are computed across the three runs. The shared output cap does not imply equal inference compute across configurations.

Cost accounting. Costs in Tables 2 and 3 are the mean estimated cost per run of the requests covering the collection's query points. We charge each request in full, including any questions outside the collection, and include recorded response attempts. Output totals include reported reasoning tokens; we apply no cache discount. Table 5 lists the fixed accounting rates used, rather than current prices or invoice totals.

Table 5: Fixed historical accounting rates in US dollars per million input or output tokens. Output includes reported reasoning tokens.

Model (mode)	Input	Output
GPT-6 Astra (high)	10.00	50.00
GPT-5.5 (high)	6.00	36.00
Qwen3.8-Max (thinking)	2.00	6.00
Gemini 3.8 Flash (thinking)	0.75	3.75
Claude Opus 5 (adaptive)	5.00	25.00
Claude Sonnet 5 (adaptive)	2.00	10.00
Kimi-K2.5 (thinking)	1.00	5.00
Kimi-K3 (thinking)	3.00	15.00
GLM-5.3 (thinking)	1.40	4.40
DeepSeek-V4-Flash (high)	1.00	3.00
Nemotron-3-Ultra (NVFP4)	0.50	2.20
Grok-4.1-fast (non-reasoning)	0.40	1.00
Llama-3.3-70B (Instruct)	1.00	1.00

B Comparison with Common Question-Type Weights

Full and Hard contain different proportions of question types. To compare performance using the same proportions, we weight each collection's type-specific accuracies by the composition of Full.

For each run, we compute $0.50A_V + 0.34A_R + 0.16A_I$, where A_V and A_R are volume and reagent-set accuracies, and A_I is indeterminate recall. The weights correspond to the 50 volume, 34 reagent-set, and 16 indeterminate-volume questions in Full. Table 6 reports the three-run means. This comparison holds the type weights fixed; it does not match the collections on protocol content or provide an independent estimate of task difficulty.

Table 6: **Accuracy with common question-type weights.** Both columns weight volume accuracy, reagent-set accuracy, and indeterminate recall by 0.50, 0.34, and 0.16, respectively. Values are means across three runs, in percent. Full is unchanged by this weighting. The final column is Full minus Hard in percentage points. These are descriptive comparisons between nested collections.

Model (mode)	Full	Hard	Difference
GPT-6 Astra (high)	64.7	32.2	32.5
GPT-5.5 (high)	64.7	30.5	34.1
Qwen3.8-Max (thinking)	61.0	30.2	30.8
Gemini 3.8 Flash (thinking)	62.0	26.0	36.0
Claude Opus 5 (adaptive)	56.0	33.6	22.4
Claude Sonnet 5 (adaptive)	53.0	33.9	19.1
Kimi-K2.5 (thinking)	55.0	28.8	26.2
Kimi-K3 (thinking)	52.0	25.4	26.6
GLM-5.3 (thinking)	42.3	15.1	27.3
DeepSeek-V4-Flash (high)	43.7	27.0	16.7
Nemotron-3-Ultra (NVFP4)	44.0	17.8	26.2
Grok-4.1-fast (non-reasoning)	33.3	29.2	4.1
Llama-3.3-70B (Instruct)	31.3	26.8	4.5

C Benchmark Comparison and Construction

Table 7 compares ProtoWorld with related benchmarks and protocol systems by workflow coverage and evaluation targets. State answerability in the table means recognizing whether the available information determines the queried state.

Table 7: Comparison of related benchmarks and protocol systems. Checks indicate explicit task capabilities, and crosses indicate no dedicated evaluation established in the reviewed task specification.

	Workflow and reasoning			**Evaluation targets**		
Benchmark / system	Real workflows	State reasoning	Quantitative reasoning	Missing information	Volume & reagent sets	State answerability
BioProBench (Liu et al., 2025)	✓	×	✓	×	×	×
PRISM (Hsu et al., 2026)	✓	✓	✓	×	×	×
ProPara (Dalvi et al., 2018)	×	✓	×	✓	×	✓
OpenPI (Tandon et al., 2020)	✓	✓	×	×	×	×
SCONE / Alchemy (Long et al., 2016)	×	✓	✓	×	×	×
QuestBench (Li et al., 2026)	×	✓	✓	✓	×	×
ClarifyBench (Suri et al., 2026)	×	✓	✓	✓	×	×
ProtoWorld	✓	✓	✓	✓	✓	✓

To illustrate how protocols become evaluation questions, Figure 4 connects the reviewed operations, reference calculations, model inputs, and scoring rules.

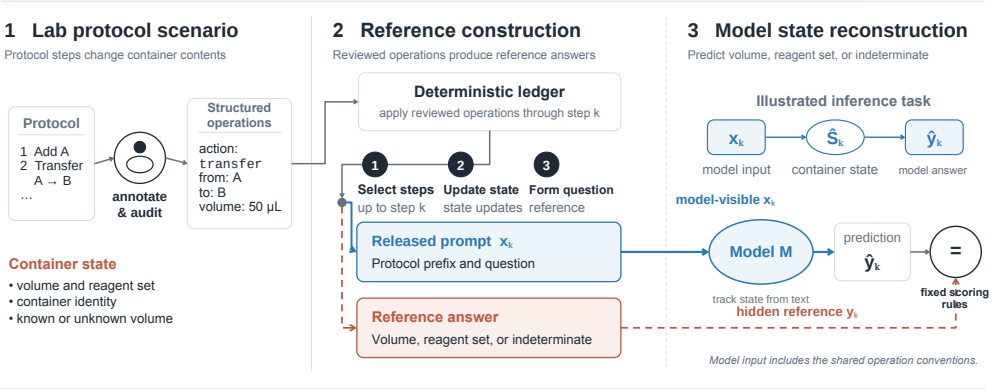


Figure 4: **Construction pipeline of ProtoWorld.** Laboratory protocols are mapped to structured operations generated by an LLM and revised by the authors. At query point k, a deterministic ledger applies the operations through that step to obtain reference answer y_k. The evaluated model M receives prompt x_k, consisting of the shared system instructions, natural-language protocol prefix, and question, and returns prediction $\hat{y}_k$. The upper-right sketch uses $\hat{S}_k$ to illustrate inferred container contents; the evaluation scores answers, not this intermediate representation. Scoring uses numerical tolerance for volumes, normalized set equality for reagent sets, and the `indeterminate` label for indeterminate-volume questions.

Protocol categories and records. The nine categories cover sequencing and genotyping, biochemical quantification, immunoassays, microbial growth, cell engineering and imaging, screening and functional assays, biosensing and molecular detection, nucleic acid extraction, and protein purification and analysis. We use these categories to organize protocols for browsing. Each source entry is a protocol record, with related protocol variants counted separately. Context length measures the text visible to a model. It differs from the number of operations that change the target container's state, examined in Appendix E.

Operation details and released records. We match different names for the same container within each protocol. When an "each well" operation produces equivalent wells, we ask a representative question. We distinguish removing a stated amount from emptying a container completely. Under the shared system prompt's rules, washing empties a well, which then contains only liquid added afterward. We retain the source text, structured operations, reference answers, and review decisions so that readers can check how each answer was obtained. The released materials also include the shared system prompt, scorer, and reagent-alias map.

Exact scoring rules. A numerical prediction $\hat{V}$ is correct when

$$|\hat{V} - V| \leq \max(0.005|V|, 10^{-6}),$$

where V is the reference volume and both volumes are expressed in μL. Reagent-set predictions require exact set equality after standardizing capitalization, whitespace, and character variants and applying the released reagent-alias map. Scoring uses neither approximate name matching nor a model judge. Missing answers, responses that cannot be parsed, and answers of the wrong type are scored as incorrect and retained in the relevant denominators. For the answerability analysis, these responses count as neither valid numerical answers nor recognized `indeterminate` responses.

D Case Study Details

These cases supplement the instruction-scope example in Section 4.2, examining which container an operation updates and how quantities and reagent contents change across later steps.

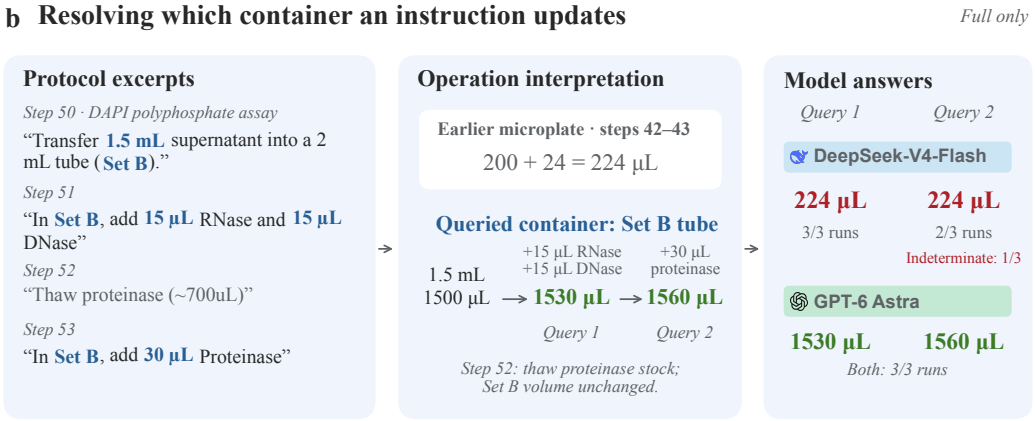


Figure 5: **Resolving which container an instruction updates.** In the DAPI-based polyphosphate assay, steps 51 and 53 add reagents to Set B, yielding 1530 and 1560 μL. Step 52 thaws proteinase stock without adding it to Set B. DeepSeek-V4-Flash's repeated 224 μL answer matches an earlier microplate recipe. DAPI is a fluorescent dye; RNase, DNase, and proteinase are enzymes. Supernatant is the liquid above settled material. Excerpts retain source wording. Colored phrases connect text to the reference interpretation; green and red denote correct and incorrect answers. Frequencies count three runs.

Updating volume and reagent contents. In a native barcoding protocol (a procedure for labeling DNA samples), a reaction contains 3.75 μL of end-prepped DNA, 1.25 μL of native barcode, and 5 μL of ligase master mix. The reference volume is 10 μL. A subsequent addition of 1 μL of the reagent EDTA increases the reference volume to 11 μL and adds EDTA to the reagent set. The volume and reagent-set questions at both query points are included in ProtoWorld Hard. Grok-4.1-fast correctly answers both the volume and reagent-set questions at the earlier query point in all three runs, but gives an incorrect volume at the later point in every run. GPT-6 Astra and GPT-5.5 return 25 μL and 26 μL at the two points in every run. Their responses capture the 1 μL increment, while both absolute volumes differ from the references. Applying the latest addition correctly is therefore insufficient if the preceding volume is wrong.

E Performance by State-Transition Depth

State-transition depth is the number of updates to a target container. We group questions by the released `n_touches` annotation into five bins: 0–1, 2, 3, 4–5, and 6+. These provisional annotations serve as a proxy for state-transition depth; the plots do not establish how many updates occurred after the latest emptying operation. Figure 6 reports accuracy on questions with determinate volume or reagent-set answers, covering 84 questions in ProtoWorld Full and 32 in ProtoWorld Hard. Figure 7 separately reports indeterminate recall on the 16 and 8 indeterminate-volume questions in the two collections. Both figures use the thirteen model configurations in Table 2.

Points show mean accuracy across three runs, and error bars show the sample standard deviation (SD) across runs. The final panel of each figure gives the number of evaluated questions in each depth bin. The answerable 6+ bin contains two questions in ProtoWorld Full and one in ProtoWorld Hard. Llama-3.3-70B answers these questions incorrectly in every run, while correctly identifying the indeterminate-volume questions in the same depth bin.

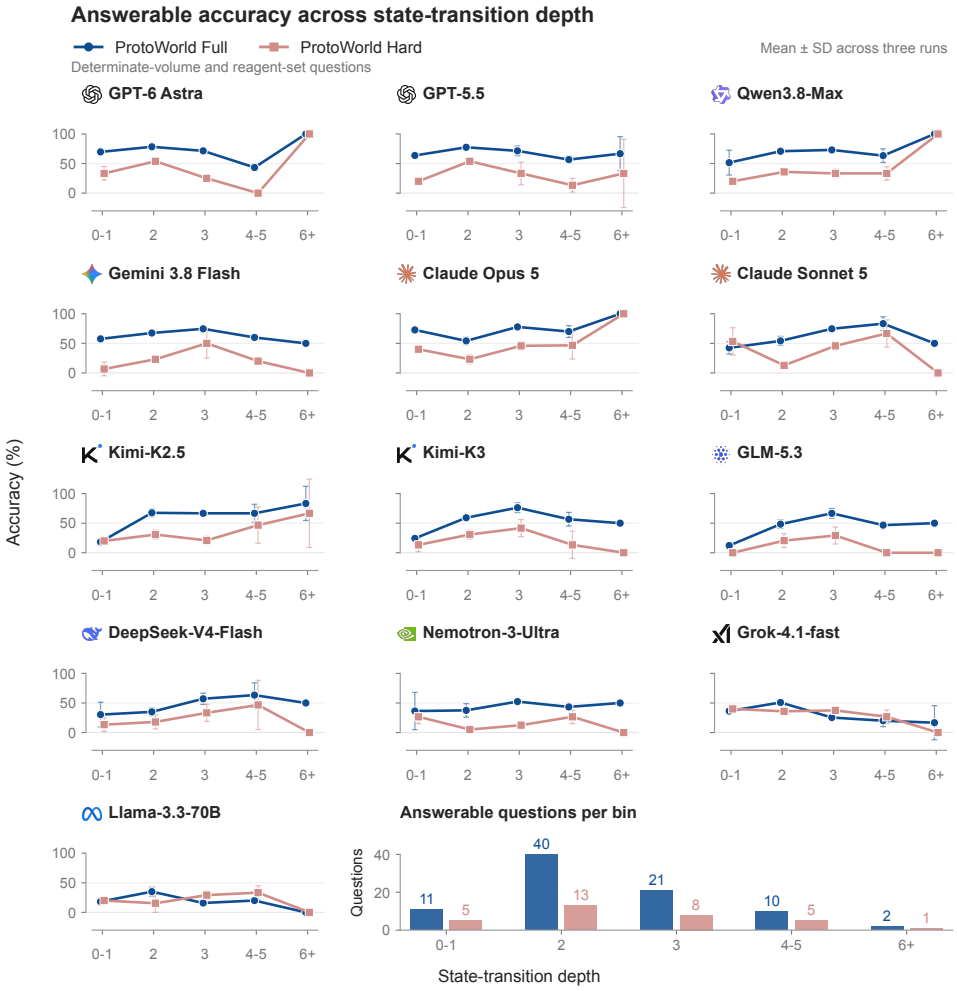


Figure 6: **Answerable accuracy across state-transition depth.** Accuracy on determinate-volume and reagent-set questions in ProtoWorld Full and ProtoWorld Hard. Each model panel shows the mean and sample standard deviation across three runs. Lines connect discrete depth bins as visual guides. The final panel reports the number of answerable questions in each bin, totaling 84 in ProtoWorld Full and 32 in ProtoWorld Hard.

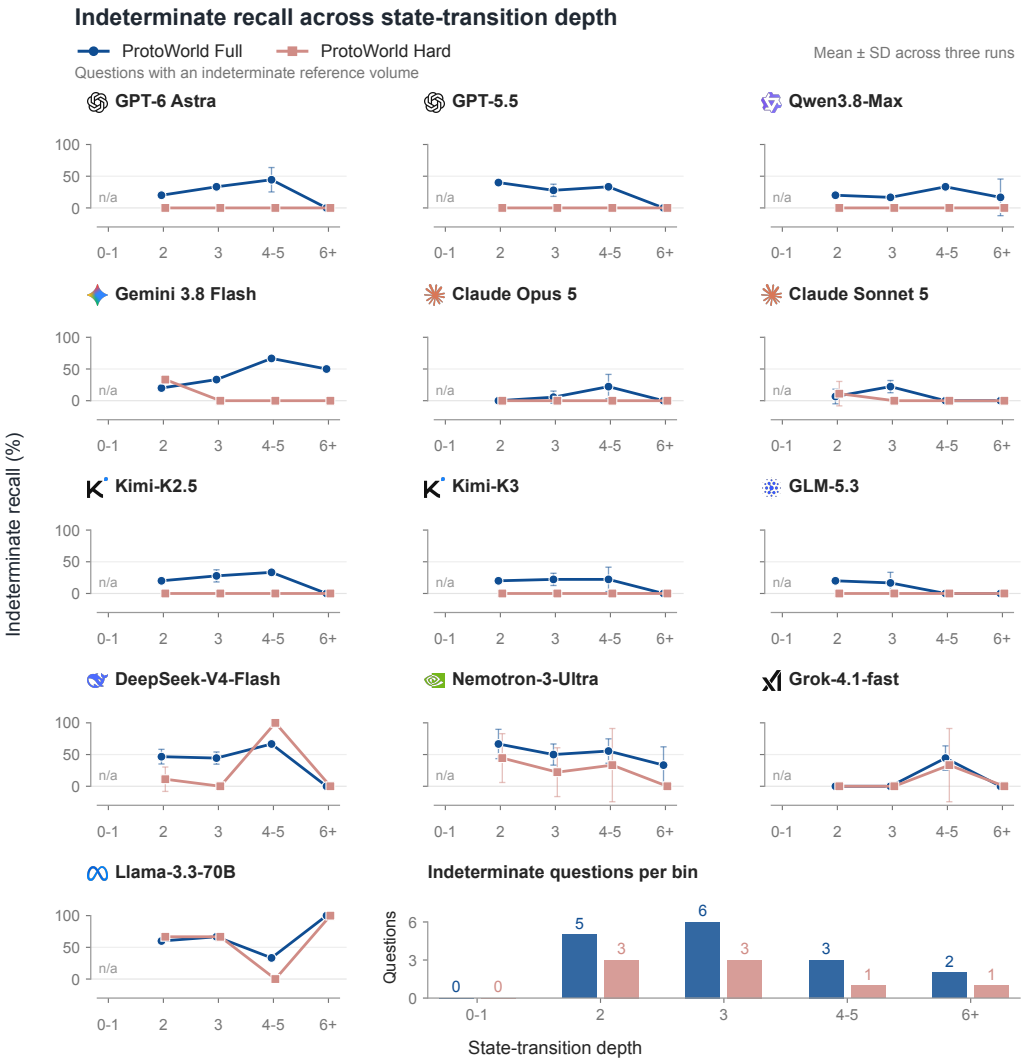


Figure 7: **Indeterminate recall across state-transition depth.** Recall on volume questions whose reference answer is `indeterminate` in ProtoWorld Full and ProtoWorld Hard. Each model panel shows the mean and sample standard deviation across three runs. Lines connect discrete depth bins as visual guides. The 0–1 bin contains no indeterminate-volume questions and is marked n/a (not applicable). The final panel reports the number of evaluated questions in each bin, totaling 16 in ProtoWorld Full and 8 in ProtoWorld Hard.

F RELIABILITY AT QUERY POINTS

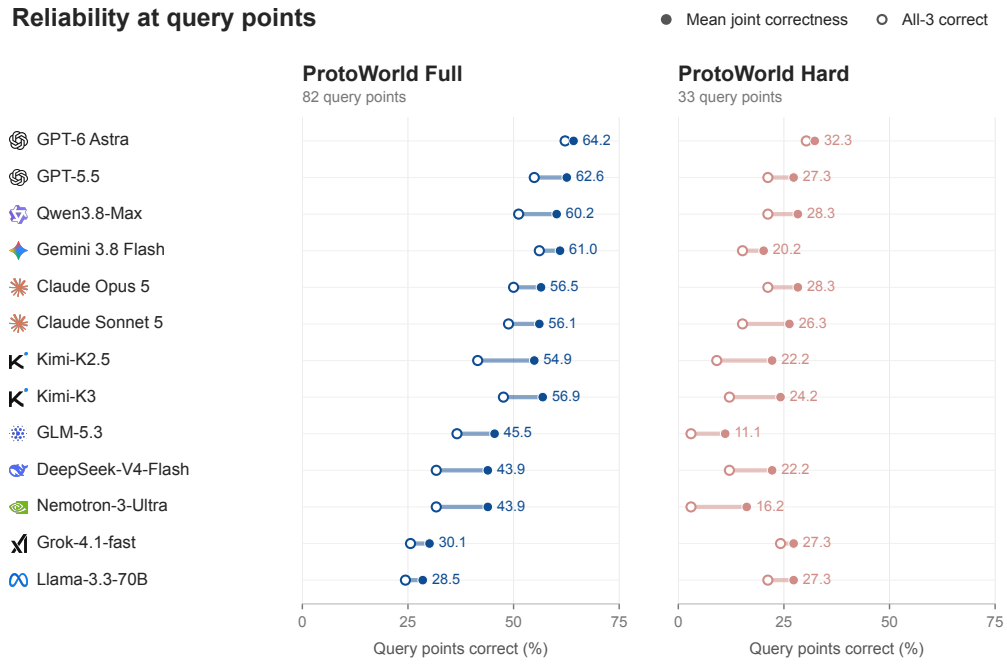


Figure 8: **Joint correctness and repeatability at query points.** A query point is correct in a run only if every question included at that point in the evaluated set is answered correctly. Filled circles show mean joint correctness across three runs, with the mean percentages printed beside them. Open circles show All-3 correct, the percentage of query points answered jointly correctly in all three runs. Connecting segments show the difference between the two metrics. Both metrics weight query points equally. The panels cover 82 query points in ProtoWorld Full and 33 in ProtoWorld Hard, using the same horizontal scale. Model order and configurations follow Table 2.