A Broader impacts

The primary goal of PAC Bench is to catalyze the development of more capable, reliable, and physically grounded VLMs and their fine-tuned variants, often called VLAs for real-world robotic applications. Because VLA fine-tuning typically relies on low-level trajectory data rather than higher level reasoning, probing the underlying VLM's understanding of object Properties, action Affordances, and physical Constraints (PAC) gives us a grounded lens into the capabilities that downstream robotic policies will inherit. By diagnosing PAC weaknesses in the base model, researchers can distinguish whether a VLA's performance stems from genuine physical common sense or simply memorized motion patterns, and thus guide targeted improvements in model architectures, training methodologies, and dataset curation. In doing so, PAC Bench helps ensure that robotic systems become more predictable, less prone to errors from a lack of physical understanding, and better equipped for safe, effective collaboration in complex, everyday environments.

By providing a fine-grained diagnostic tool, PAC Bench can help researchers and developers identify specific weaknesses in current models, thereby guiding targeted improvements in model architectures, training methodologies, and dataset curation. This, in turn, can lead to robotic systems that are more predictable, less prone to errors stemming from a lack of physical common sense, and better able to perform a wide range of useful tasks. The open release of our benchmark and its diverse data sources (including web-scale images, real-world humanoid captures, and simulated scenarios) is intended to foster broad community engagement and accelerate progress in this crucial area of AI.

While any advancement in AI capabilities warrants ongoing consideration of its societal implications, our work focuses on enhancing the fundamental understanding and robustness of AI systems, which we see as a positive step towards more responsible AI development. We encourage the community to leverage PAC Bench to build systems that not only demonstrate impressive capabilities but also operate with a clear and verifiable understanding of their physical environment, ultimately contributing to the beneficial integration of AI into society.

B Experimental Setup

This appendix provides further details on the experimental setup used for collecting data and for evaluating VLMs on PAC Bench, complementing Section 4.1 of the main paper.

B.1 Models Evaluated and Access

The VLM evaluations reported in this paper (Section 4) encompass a diverse suite of models. All models were accessed via their respective APIs available through the OpenRouter service[2] between April 2024 and May 2024. The specific models evaluated are detailed below, along with their OpenRouter paths:

1. **Claude 3.7 Sonnet:** `https://openrouter.ai/anthropic/claude-3.7-sonnet`
2. **Claude 3.7 Sonnet (T):** `https://openrouter.ai/anthropic/claude-3.7-sonnet:thinking` *(This denotes Chain-of-Thought prompting applied to the Claude 3.7 Sonnet model.)*
3. **Claude 3.5 Sonnet:** `https://openrouter.ai/anthropic/claude-3.5-sonnet`
4. **Gemini 2.0 Flash 001:** `https://openrouter.ai/google/gemini-2.0-flash-001`
5. **Gemini 2.5 Flash P:** `https://openrouter.ai/google/gemini-2.5-flash-preview`
6. **Gemini 2.5 Pro P:** `https://openrouter.ai/google/gemini-2.5-pro-preview-03-25`
7. **GPT-4.1:** `https://openrouter.ai/openai/gpt-4.1`
8. **o4-mini-high:** `https://openrouter.ai/openai/o4-mini-high` *(Note: The "(T)" for this model in some tables also indicates Chain-of-Thought prompting.)*
9. **GPT-4.1 Mini:** `https://openrouter.ai/openai/gpt-4.1-mini`

[2] `https://openrouter.ai/`

10. **Llama 4 Maverick:** `https://openrouter.ai/meta-llama/llama-4-maverick`
11. **Llama 4 Scout:** `https://openrouter.ai/meta-llama/llama-4-scout`
12. **Llama 3.2 90B VI:** `https://openrouter.ai/meta-llama/llama-3.2-90b-vision-instruct` *(VI denotes Vision Instruct. Your tables may use Llama 3.2 90B Vision I)*
13. **Grok 2 Vision:** `https://openrouter.ai/x-ai/grok-2-vision-1212`
14. **Grok Vision Beta:** `https://openrouter.ai/x-ai/grok-vision-beta`
15. **Qwen2.5 VL:** `https://openrouter.ai/qwen/qwen2.5-vl-72b-instruct` *(VL denotes Vision Language.)*
16. **Qwen VL Plus:** `https://openrouter.ai/qwen/qwen-vl-plus`
17. **Qwen 3 (235B):** `https://openrouter.ai/qwen/qwen3-235b-a22b` *(This appears as "Qwen 3".)*

B.2 Simulated Constraint Scenario Generation

To generate a diverse and controllable set of scenarios for evaluating VLM understanding of physical constraints, we developed a simulation-based pipeline using the MuJoCo physics engine. This approach allows for the systematic creation of situations where specific physical limitations are the primary factor determining task feasibility. Our design focused on four primary constraint domains critical for robotic manipulation:

Figure 6: Example scenes corresponding to each constraint domain (left to right): (a) *Impossible Placement*: attempting to fit the blue block into the brown box; (b) *Occlusion/Support Issues*: picking up the red block beneath another; (c) *Stability Constraints*: lifting the unstable green block from the top of a stack; (d) *Reachability and Access Constraints*: grasping a block placed at the very edge of the workspace.

- **Impossible Placement:** These scenarios test the understanding of basic stability and geometric fit. Examples include attempting to place a larger object into a smaller container or balancing an object on an inadequately small or unstable surface. Objects with varying shapes (e.g., cubes, spheres, custom meshes) and sizes were utilized.

  Prompt: "I want to place the blue object inside the brown box, are there any constraints that prevent me from doing so or is it possible?"
  Verification Prompt: "Does the following text say that the task is impossible? Answer only with 'true' or 'false'."

- **Occlusion/Support Issues:** These scenarios evaluate reasoning about object interdependencies and access. A typical setup involves a target object being either partially or fully occluded by another object, or where the target object is currently supporting another, making its removal problematic (e.g., preventing a stack from toppling if a lower block is removed).

  Prompt: "I want to pick up the red object from the bottom of the stack, are there any constraints that prevent me from doing so or is it possible?"
  Verification Prompt: "Does the following text say that object on the top might fall due to it's placement? Answer only with 'true' or 'false'."

- **Stability Constraints:** These focus on the inherent stability of an object or an assembly if an action is performed. Examples include attempting to pick a block from an unstable stack where the act of picking itself or the removal of the object leads to the collapse of the

Figure 7: Samples from robocasa datapoint in PACBench

remaining structure, or attempting to place an object such that the resulting configuration is unstable.

> *Prompt:* "I want to pick up the green object from the top of the stack, are there any constraints that prevent me from doing so or is it possible?"
> *Verification Prompt:* "Does the following text say that object on the top might fall due to it's placement? Answer only with 'true' or 'false'."

- **Reachability and Access Constraints:** These scenarios test understanding of spatial and kinematic limitations. Objects might be placed at the edge of a workspace, behind obstacles, or in orientations that make them difficult or impossible for a standard robotic gripper to access without collision or exceeding plausible joint limits.

  > *Prompt:* "I want to pick up the red object from the edge of the stack, are there any constraints that prevent me from doing so or is it possible?"
  > *Verification Prompt:* "Does the following text say that object is out of reach? Answer only with 'true' or 'false'."

For each of these four domains, we procedurally generated **10 distinct environment instantiations**. Randomization was applied to object properties (e.g., slight variations in size and mass where relevant for dynamics), initial positions and orientations, as well as the placement of minor distractor objects to increase visual diversity while ensuring the core constraint remained salient.

Figure 6 provides a visual summary of one example from each sub-domain.

B.3 Synthetic Object-Centric Dataset from RoboCasa Assets

To support fine-grained object reasoning evaluations, we constructed a synthetic image dataset by curating a subset of authentic 3D meshes from the RoboCasa simulation framework. While RoboCasa provides a rich large-scale kitchen environment with hundreds of AI-generated and hand-modeled assets, we selected only the 45 objects that had artist-modeled meshes (i.e., excluding purely AI-generated models). Each object is paired with high-resolution renders, manual affordance annotations, and detailed physical/property labels.

- **Asset Selection:** We chose 45 common kitchen and tabletop items, spanning foodstuffs, containers, utensils, and small appliances. The full set is: `apple`, `baguette`, `beer`, `bottled_water`, `bowl`, `boxed_food`, `broccoli`, `candle`, `cereal`, `cheese`, `chocolate`, `corn`, `croissant`, `cucumber`, `cupcake`, `cutting_board`, `donut`, `egg`, `eggplant`, `jug`, `ketchup`, `kettle_non_electric`, `knife`, `lime`, `liquor`, `milk`, `onion`, `orange`, `pan`, `peach`, `pot`, `potato`, `shaker`, `spatula`, `sponge`, `spoon`, `spray`, `sweet_potato`, `tangerine`, `teapot`, `tomato`, `tray`, `waffle`, `wine`, `yogurt`.
- **Viewpoint Sampling:** Each object was rendered from **24** distinct viewpoints by rotating the camera around the object's vertical axis (Z) at three elevations ($-30°, 0°, +30°$) and eight azimuths ($0°, 45°, \ldots, 315°$). Filenames follow the pattern:

  `elev<elevation>_azim<azimuth>.png`

  for example `elev-30_azim135.png`, yielding $45 \times 24 = 1080$ high-resolution images.
- **Affordance Annotation and Evaluation:** We hand-annotated **41** of the 45 objects with one or more affordances (e.g., *edible*, *pourable*, *stackable*). To probe model understanding, we used the prompt:

  ```
  List all the possible affordances of a <object_name>.
  An affordance is what an object can be used for or what
  actions can be performed with it.  List them in a clear,
  comma-separated format.
  ```

  We then computed two strict metrics:

  1. **All-correct:** Does the LLM output contain *all* ground-truth affordances?
  2. **At-least-one:** Does the LLM output contain *at least one* ground-truth affordance?

  Verification prompts were:

  ```
  Given the following ground truth affordances for a
  <object_name>:  <list>
  And the following LLM response:  <llm_response>
  Does the LLM response contain all the ground truth
  affordances?  Answer only with 'true' or 'false'.
  ```

  ```
  Given the following ground truth affordances for a
  <object_name>:  <list>
  And the following LLM response:  <llm_response>
  Does the LLM response contain at least one of the ground
  truth affordances?  Answer only with 'true' or 'false'.
  ```
- **Property Annotation and Evaluation:** We manually labeled each object with up to 11 physical and functional properties: `COLOR`, `COMPLEXITY`, `CONSUMABILITY`, `DENSITY`, `HARDNESS`, `STICKINESS`, `THICKNESS`, `WEIGHT`, `CAPACITY`, `CONTENTS`, and `SEALING`. Table 5 summarizes the number of objects annotated per property. For example, *yogurt* was annotated as:

  ```
  yogurt|WEIGHT|Medium|Moderate, Balanced
  yogurt|COLOR|Multicolored|Gradient, Striped
  yogurt|HARDNESS|Hard|Solid, Rigid
  ...
  ```

  Each property uses a predefined set of discrete options and synonyms. We defined:

  ```
  WEIGHT_options = """
  Light: Featherweight, Lightweight
  Medium: Moderate, Balanced
  Heavy: Bulky, Dense
  Dynamic: Fluctuating, Variable
  """

  COLOR_options = """
  Monochromatic: Single Color, Neutral
  ```

```
Multicolored: Gradient, Striped
Metallic: Glossy, Shiny
Matte: Flat, Dull
"""

HARDNESS_options = """
Hard: Solid, Rigid
Soft: Plush, Flexible
Brittle: Fragile, Breakable
"""

ORIENTATION_options = """
Vertical: Upright, Standing
Horizontal: Flat, Reclined
Multi-directional: Rotational, Adjustable
"""

CONSUMABILITY_options = """
Consumable: Edible, Burnable, Disposable
Non-consumable: Reusable, Permanent
"""

COMPLEXITY_options = """
Simple: Single-unit, Monolithic
Multi-object: Assembled, Interconnected
"""

CAPACITY_options = """
Containable: Hollow, Enclosable
Non-containable: Solid, Unperforated
"""

CONTENTS_options = """
Contains: Filled, Occupied
Empty: Vacant, Void
"""

SEALING_options = """
Sealed: Airtight, Watertight
Unsealed: Open, can leak
"""

DENSITY_options = """
High-density: Dense, Compact
Low-density: Lightweight, Buoyant
Variable: Adjustable, Fluid
"""

THICKNESS_options = """
Thin: Slim, Minimal Thickness
Medium: Standard Thickness, Balanced
Thick: Sturdy, Bulky
"""

STICKINESS_options = """
Sticky: Adhesive, Tacky
Non-sticky: Smooth, Slippery
Variable: Temporary Stickiness, Conditional Adhesion
"""
```

Models were queried with the following template:

```
Evaluate the {property} of the object(s) enclosed within the
red bounding box in the image.
Respond with only one of the following options:
{options}
Provide no additional text, explanations, or numbers.
```

Property	# Objects Annotated
COLOR	41
COMPLEXITY	41
CONSUMABILITY	41
DENSITY	41
HARDNESS	41
STICKINESS	41
THICKNESS	41
WEIGHT	41
CAPACITY	39
CONTENTS	38
SEALING	19

Table 5: Number of objects annotated per property.

Overall, this dataset comprises **1080** images of 45 objects, enriched with manual affordance and property labels, enabling comprehensive evaluation of VLM performance on view-invariant recognition, affordance inference, and property classification tasks.

B.4 Open Images V7 Subset for Object-Centric Affordance and Property Evaluation

Open Images V7 is a comprehensive, real-world image corpus of approximately 1.9 million images spanning 600 object classes, annotated with image-level labels, bounding boxes, segmentation masks, visual relationships, and localized narratives. From this large-scale dataset, we selected 116 object classes for which single-instance examples could be clearly isolated and annotated. For each class, we sampled between four and eight representative images, yielding a total of 679 unique frames. Filenames conform to the pattern `<object_id>_<image_id>.jpg` (e.g. `012w5l_226957c99fab6ddf.jpg`), where the first token denotes the Open Images class identifier and the second is the image hash. In every image, exactly one instance of the target object is marked with a yellow bounding box (see Fig. 8).

To probe visual-language models' understanding of object affordances, we gathered human annotations at the class level, specifying between one and three affordances per object (for example, "Sit", "Pour", or "Cut"). These annotations were recorded in CSV form as `object,affordance1,affordance2,affordance3`, resulting in over 300 total affordance entries across the 116 classes. Model outputs are evaluated under two strict criteria: (1) whether all ground-truth affordances appear in the response ("all-correct"), and (2) whether at least one ground-truth affordance appears ("at-least-one"). Verification is automated via prompts that present the ground-truth list alongside the model's response and request a single answer of "true" or "false."

In addition to affordances, we annotated each image for up to 15 physical and functional properties (COLOR, COMPLEXITY, CONSUMABILITY, DENSITY, HARDNESS, STICKINESS, THICKNESS, WEIGHT, CAPACITY, CONTENTS, SEALING, ORIENTATION, plus four domain-specific traits). Over 12,421 annotation entries were collected, corresponding to 10,506 unique (image, property) pairs—some images received multiple annotations for the same property. The distribution of annotations per property file is summarized below:

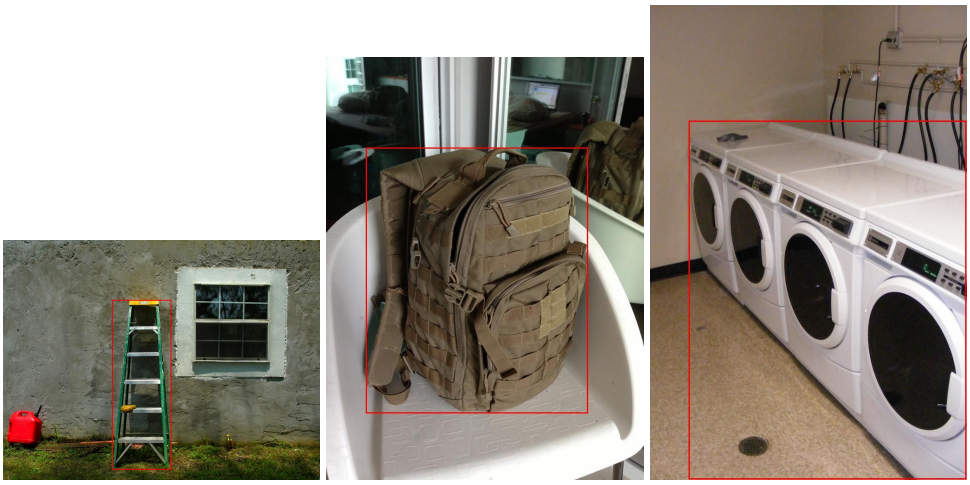

Figure 8: Example from our Open Images subset: a single object annotated with a red bounding box.

Property File	Lines
property_CAPACITY_.csv	679
property_COLOR_.csv	818
property_COMPLEXITY_.csv	1140
property_CONSUMABILITY_.csv	679
property_CONTENTS_.csv	679
property_DENSITY_.csv	679
property_HARDNESS_.csv	679
property_ORIENTATION_.csv	887
property_SEALING_.csv	871
property_STICKINESS_.csv	1358
property_THICKNESS_.csv	679
property_WEIGHT_.csv	1358

Models are queried with the template:

```
Evaluate the {property} of the object(s) enclosed within the
red bounding box in the image.
Respond with only one of the following options:  {options}
Provide no additional text, explanations, or numbers.
```

Because Open Images V7 comprises 600 classes and nearly two million images, this protocol can be extended seamlessly to new categories and additional examples. Once class-level affordance and property labels are established, any further images sampled under the same class identifier inherit those annotations, enabling scalable evaluation of view-invariant recognition, affordance inference, and physical attribute classification.

B.5 Embodied Robot Capture: Unitree G1 Dual-Arm Dataset

To complement our web-sourced and simulated resources with truly embodied visual data, we collected a fresh corpus of interactions using a dual-arm Unitree G1 humanoid operating in an indoor laboratory. The robot was tele-operated or executed short, pre-programmed primitives at a standing workstation filled with diverse household objects that were *not* present in either our RoboCasa or Open-Images subsets, thereby increasing inter-dataset heterogeneity. Each scene was photographed simultaneously from two calibrated perspectives: an egocentric camera rigidly attached to the robot's head (1280×720 at 30 Hz) and a side-mounted static camera that offered a wider allocentric view of the workspace. The resulting paired images allow Vision–Language Models (VLMs) to be probed under both first- and third-person viewpoints—conditions that often lead to markedly different perceptual challenges in robotics.

Figure 9: Samples from Unitree G1 humanoid from PacBench

Property annotations. For every object-centric tabletop configuration we recorded up to twelve physical and functional properties using the controlled vocabulary introduced in previous sections (e.g., `WEIGHT`, `COLOR`, `SEALING`). A total of 785 property rows were produced across 67 unique image pairs, giving an average of roughly twelve properties per scenario. All properties except `SEALING` are exhaustively annotated for every scene; `SEALING` appears in 48 of the 67 cases, reflecting either inapplicability or annotator uncertainty for the remaining scenes. Distributions are well balanced: for example, the `WEIGHT` axis splits into *Light* (49 %), *Medium* (42 %), and *Heavy* (9 %), while `COLOR` is almost evenly divided between *Monochromatic* and *Multicolored* with a small metallic tail. Descriptor-level statistics show that every categorical choice is accompanied by its canonical pair of synonyms (e.g., *Dense, Compact* whenever *High-density* is selected), a consequence of the structured drop-down interface used during labelling.

Affordance annotations. Sixty-eight scenarios were further enriched with up to three free-form affordances per object, resulting in 181 individual affordance strings. Half of the scenes list a full triplet, roughly 43 % include two entries, and only seven per cent contain a single affordance. The vocabulary is intentionally open; nevertheless several patterns emerge—"act as weight" accounts for 18 % of all mentions, followed by "contain things" and "scrape things." Frequent combinations such as *(contain things, act as cushion, act as weight)* illustrate that annotators naturally link physical support, compliance, and mass when reasoning about everyday artefacts. Evaluation uses the same strict "all-correct" and "at-least-one" metrics adopted for our other datasets, coupled with the verification prompts described earlier.

Constraint annotations. Finally, 53 of the scenarios include a natural-language question about the feasibility of a specific robot action together with a short justification when the answer is negative. These queries test spatial reasoning (e.g., balancing a cube on a pyramid), containment under orientation changes (placing items inside an inverted pen-stand), and accessibility issues (writing when a marker cap is closed). Recurrent keywords such as *inverted*, *balance*, *upright*, and *closed* reveal the dominant failure modes considered. Although the majority of responses start with a terse "No," the accompanying explanations provide fine-grained cues that are invaluable for evaluating whether a VLM can pinpoint the exact limiting factor.

Cross-modal linking and usage. Because every record—whether property, affordance, or constraint—references the same `cam0_file` / `cam1_file` pair, researchers can seamlessly join the three ground-truth tables to obtain a fully articulated description of each physical scene. This makes it possible to explore, for instance, how an object's annotated orientation (`Vertical`, `Horizontal`, `Multi-directional`) influences both its perceived affordances and the constraints imposed on manipulation tasks. The corpus therefore serves as a high-fidelity test-bed for embodied VLM evaluation, filling the gap between purely synthetic renders and images scraped from the web. In total, the Unitree G1 set delivers 67–68 richly annotated scenarios, amounting to hundreds of individual labels that capture the intertwined facets of Properties, Affordances, and Constraints from a truly robot-centric vantage point.

B.6 Computational Resource

Evaluations were conducted by querying their respective publicly available APIs from OpenRouter[3]. Due to the nature of API access, precise underlying hardware details are not available for these models, and performance can be subject to API latency and load. Estimated API costs for some initial property evaluations were a factor in scoping the experiments, as noted in Section 3.1 regarding the exclusion of the RoboCasa image set from the current VLM evaluation suite.

The total cost for running each model for all reported results in the main paper is as follows:

1. **Claude 3.7 Sonnet:** 108.5$
2. **Claude 3.7 Sonnet (T):** 167.8$
3. **Claude 3.5 Sonnet:** 73.9$
4. **Gemini 2.0 Flash 001:** 2.6$
5. **Gemini 2.5 Flash P:** 2.9$
6. **Gemini 2.5 Pro P:** 150.2$
7. **GPT-4.1:** 25.9$
8. **o4-mini-high:** 48.0$
9. **GPT-4.1 Mini:** 5.5$
10. **Llama 4 Maverick:** 40.8$
11. **Llama 4 Scout:** 2.2$
12. **Llama 3.2 90B VI:** 16.8$
13. **Grok 2 Vision:** 66.7$
14. **Grok Vision Beta:** 22.4$
15. **Qwen2.5 VL:** 8.7$
16. **Qwen VL Plus:** 2.4$
17. **Qwen 3 (235B):** 24.0$

Overall Cost Summary

The total estimated cost for running all models across the entire PAC benchmark is $769.30. The cost breakdown by PAC category, aggregated across all models, is as follows:

- Properties: $695.76
- Affordances: $62.31
- Constraints: $11.23

Detailed Cost Breakdown by Dataset (Aggregated Across All Models)

The costs, aggregated across all models but broken down by individual datasets within each PAC category, are:

- Properties - Real Robot: $93.89
- Properties - Open Images: $601.87
- Affordances - Real Robot: $8.24
- Affordances - Open Images: $54.07
- Constraints - Mujoco: $4.78
- Constraints - Real Robot: $6.45

[3] https://openrouter.ai/

Model-Specific Cost Breakdown

This section details the estimated cost for each model, distributed across the PAC categories and individual datasets. These costs are derived by proportionally distributing the total category/dataset costs based on each model's normalized cost relative to the sum of all normalized model costs (where normalization is performed against the least expensive model, `meta-llama/llama-4-scout`, which has a raw cost of $2.20).

Costs for `anthropic/claude-3.7-sonnet` **PAC Category Costs:**

- Properties: $98.13
- Affordances: $8.79
- Constraints: $1.58

Individual Dataset Costs:

- Properties - Real Robot: $13.24
- Properties - Open Images: $84.89
- Affordances - Real Robot: $1.16
- Affordances - Open Images: $7.63
- Constraints - Mujoco: $0.67
- Constraints - Real Robot: $0.91

Costs for `anthropic/claude-3.7-sonnet:thinking` **PAC Category Costs:**

- Properties: $151.75
- Affordances: $13.59
- Constraints: $2.45

Individual Dataset Costs:

- Properties - Real Robot: $20.48
- Properties - Open Images: $131.28
- Affordances - Real Robot: $1.80
- Affordances - Open Images: $11.79
- Constraints - Mujoco: $1.04
- Constraints - Real Robot: $1.41

Costs for `anthropic/claude-3.5-sonnet` **PAC Category Costs:**

- Properties: $66.83
- Affordances: $5.99
- Constraints: $1.08

Individual Dataset Costs:

- Properties - Real Robot: $9.02
- Properties - Open Images: $57.82
- Affordances - Real Robot: $0.79
- Affordances - Open Images: $5.19
- Constraints - Mujoco: $0.46
- Constraints - Real Robot: $0.62

Costs for `google/gemini-2.0-flash-001` **PAC Category Costs:**

- Properties: $2.35
- Affordances: $0.21
- Constraints: $0.04

Individual Dataset Costs:

- Properties - Real Robot: $0.32
- Properties - Open Images: $2.03
- Affordances - Real Robot: $0.03
- Affordances - Open Images: $0.18
- Constraints - Mujoco: $0.02
- Constraints - Real Robot: $0.02

Costs for `google/gemini-2.5-flash-preview` **PAC Category Costs:**

- Properties: $2.63
- Affordances: $0.24
- Constraints: $0.04

Individual Dataset Costs:

- Properties - Real Robot: $0.35
- Properties - Open Images: $2.27
- Affordances - Real Robot: $0.03
- Affordances - Open Images: $0.20
- Constraints - Mujoco: $0.02
- Constraints - Real Robot: $0.02

Costs for `google/gemini-2.5-pro-preview-03-25` **PAC Category Costs:**

- Properties: $135.84
- Affordances: $12.17
- Constraints: $2.19

Individual Dataset Costs:

- Properties - Real Robot: $18.33
- Properties - Open Images: $117.51
- Affordances - Real Robot: $1.61
- Affordances - Open Images: $10.56
- Constraints - Mujoco: $0.93
- Constraints - Real Robot: $1.26

Costs for `openai/gpt-4.1` **PAC Category Costs:**

- Properties: $23.42
- Affordances: $2.10
- Constraints: $0.38

Individual Dataset Costs:

- Properties - Real Robot: $3.16
- Properties - Open Images: $20.26
- Affordances - Real Robot: $0.28
- Affordances - Open Images: $1.82
- Constraints - Mujoco: $0.16
- Constraints - Real Robot: $0.22

Costs for `openai/o4-mini-high` **PAC Category Costs:**

- Properties: $43.42
- Affordances: $3.89
- Constraints: $0.70

Individual Dataset Costs:

- Properties - Real Robot: $5.86
- Properties - Open Images: $37.56
- Affordances - Real Robot: $0.51
- Affordances - Open Images: $3.37
- Constraints - Mujoco: $0.30
- Constraints - Real Robot: $0.40

Costs for `openai/gpt-4.1-mini` **PAC Category Costs:**

- Properties: $4.97
- Affordances: $0.45
- Constraints: $0.08

Individual Dataset Costs:

- Properties - Real Robot: $0.67
- Properties - Open Images: $4.30
- Affordances - Real Robot: $0.06
- Affordances - Open Images: $0.39
- Constraints - Mujoco: $0.03
- Constraints - Real Robot: $0.05

Costs for `meta-llama/llama-4-maverick` **PAC Category Costs:**

- Properties: $36.91
- Affordances: $3.31
- Constraints: $0.60

Individual Dataset Costs:

- Properties - Real Robot: $4.98
- Properties - Open Images: $31.93
- Affordances - Real Robot: $0.44
- Affordances - Open Images: $2.87
- Constraints - Mujoco: $0.25
- Constraints - Real Robot: $0.34

Costs for `meta-llama/llama-4-scout` **PAC Category Costs:**

- Properties: $1.99
- Affordances: $0.18
- Constraints: $0.03

Individual Dataset Costs:

- Properties - Real Robot: $0.27
- Properties - Open Images: $1.72
- Affordances - Real Robot: $0.02
- Affordances - Open Images: $0.15
- Constraints - Mujoco: $0.01
- Constraints - Real Robot: $0.02

Costs for `meta-llama/llama-3.2-90b-vision-instruct` **PAC Category Costs:**

- Properties: $15.20
- Affordances: $1.36
- Constraints: $0.25

Individual Dataset Costs:

- Properties - Real Robot: $2.05
- Properties - Open Images: $13.15
- Affordances - Real Robot: $0.18
- Affordances - Open Images: $1.18
- Constraints - Mujoco: $0.10
- Constraints - Real Robot: $0.14

Costs for `x-ai/grok-2-vision-1212` **PAC Category Costs:**

- Properties: $60.33
- Affordances: $5.40
- Constraints: $0.97

Individual Dataset Costs:

- Properties - Real Robot: $8.14
- Properties - Open Images: $52.19
- Affordances - Real Robot: $0.71
- Affordances - Open Images: $4.69
- Constraints - Mujoco: $0.41
- Constraints - Real Robot: $0.56

Costs for `x-ai/grok-vision-beta` **PAC Category Costs:**

- Properties: $20.26
- Affordances: $1.81
- Constraints: $0.33

Individual Dataset Costs:

- Properties - Real Robot: $2.73
- Properties - Open Images: $17.52
- Affordances - Real Robot: $0.24
- Affordances - Open Images: $1.57
- Constraints - Mujoco: $0.14
- Constraints - Real Robot: $0.19

Costs for `qwen/qwen2.5-vl-72b-instruct` **PAC Category Costs:**

- Properties: $7.86
- Affordances: $0.70
- Constraints: $0.13

Individual Dataset Costs:

- Properties - Real Robot: $1.06
- Properties - Open Images: $6.80
- Affordances - Real Robot: $0.09
- Affordances - Open Images: $0.61
- Constraints - Mujoco: $0.05
- Constraints - Real Robot: $0.07

Costs for `qwen/qwen-vl-plus` **PAC Category Costs:**

- Properties: $2.17
- Affordances: $0.19
- Constraints: $0.04

Individual Dataset Costs:

- Properties - Real Robot: $0.29
- Properties - Open Images: $1.88
- Affordances - Real Robot: $0.03
- Affordances - Open Images: $0.17
- Constraints - Mujoco: $0.01
- Constraints - Real Robot: $0.02

Costs for `qwen/qwen3-235b-a22b` **PAC Category Costs:**

- Properties: $21.71
- Affordances: $1.94
- Constraints: $0.35

Individual Dataset Costs:

- Properties - Real Robot: $2.93
- Properties - Open Images: $18.78
- Affordances - Real Robot: $0.26
- Affordances - Open Images: $1.69
- Constraints - Mujoco: $0.15
- Constraints - Real Robot: $0.20

C Dataset Statistics

C.1 Properties

Real Robo

The *Real Robo* properties subset contains 785 annotations spread across 67 unique scenario image–pairs, giving a mean of 11.7 annotated properties per scenario (the schema expects 12).

Property–name frequency. Every property except SEALING appears exactly 67 times, corresponding to 8.54 % of all annotations each. SEALING appears 48 times (6.11 %).

Category distribution (overall). Non-consumable 67 (8.54 %), Medium thickness 63 (8.03 %), Non-sticky 55 (7.01 %), Contains 50 (6.37 %), Non-containable 38 (4.84 %), Horizontal 38 (4.84 %), Hard 36 (4.59 %), Simple 36 (4.59 %), High-density 34 (4.33 %), Light 33 (4.20 %), Multicolored 33 (4.20 %), Low-density 33 (4.20 %), Soft 31 (3.95 %), Multi-object 31 (3.95 %), Sealed 29 (3.69 %), Containable 29 (3.69 %), Monochromatic 29 (3.69 %), Unsealed 19 (2.42 %), Vertical 19 (2.42 %), Empty 17 (2.17 %), Thick 16 (2.04 %), Thin 16 (2.04 %), Multi-directional 10 (1.27 %), Sticky 7 (0.89 %), Heavy 6 (0.76 %), Metallic 5 (0.64 %), Variable 5 (0.64 %).

Category distribution per property. CAPACITY: Non-containable 38 (56.7 %), Containable 29 (43.3 %). COLOR: Multicolored 33 (49.3 %), Monochromatic 29 (43.3 %), Metallic 5 (7.5 %). COMPLEXITY: Simple 36 (53.7 %), Multi-object 31 (46.3 %). CONSUMABILITY: Non-consumable 67 (100 %). CONTENTS: Contains 50 (74.6 %), Empty 17 (25.4 %). DENSITY: High-density 34 (50.8 %), Low-density 33 (49.2 %). HARDNESS: Hard 36 (53.7 %), Soft 31 (46.3 %). ORIENTATION: Horizontal 38 (56.7 %), Vertical 19 (28.4 %), Multi-directional 10 (14.9 %). SEALING: Sealed 29 (60.4 %), Unsealed 19 (39.6 %). STICKINESS: Non-sticky 55 (82.1 %), Sticky 7 (10.4 %), Variable 5 (7.5 %). THICKNESS: Medium 35 (52.2 %), Thick 16 (23.9 %), Thin 16 (23.9 %). WEIGHT: Light 33 (49.3 %), Medium 28 (41.8 %), Heavy 6 (9.0 %).

Descriptor distribution (overall). Solid 74 (4.71 %); Reusable 67, Permanent 67 (4.27 % each); Lightweight 66 (4.20 %); Balanced 63 (4.01 %); Smooth 55, Slippery 55 (3.50 % each); Filled 50, Occupied 50 (3.18 % each); Dense 40 (2.55 %); Flat 38, Reclined 38, Unperforated 38 (2.42 % each); Rigid 36, Single-unit 36, Monolithic 36 (2.29 % each); Standard Thickness 35 (2.23 %); Compact 34 (2.17 %); Gradient 33, Striped 33, Featherweight 33, Buoyant 33 (2.10 % each); Assembled 31, Interconnected 31, Plush 31, Flexible 31 (1.97 % each); Airtight 29, Watertight 29, Single Color 29, Neutral 29, Hollow 29, Enclosable 29 (1.85 % each); Moderate 28 (1.78 %); Bulky 22 (1.40 %); Upright 19, Standing 19, Open 19, Can-leak 19 (1.21 % each); Vacant 17, Void 17 (1.08 % each); Sturdy 16, Slim 16, Minimal Thickness 16 (1.02 % each); Rotational 10, Adjustable 10 (0.64 % each); Adhesive 7, Tacky 7 (0.45 % each); Glossy 5, Shiny 5, Temporary Stickiness 5, Conditional Adhesion 5 (0.32 % each).

Descriptor distribution per property & category. Each category listed above is characterised by exactly two descriptors, each accounting for half of the annotations in that category—for example, Containable objects are equally annotated as *Hollow* and *Enclosable*, Metallic objects as *Glossy* and *Shiny*, Hard objects as *Solid* and *Rigid*, and so on across all 27 property–category pairs.

Robocasa

The *Robocasa* synthetic-properties subset comprises 424 property annotations describing 41 distinct household objects (≈10.3 properties per object).

Property–name frequency. Eight properties (WEIGHT, COLOR, HARDNESS, CONSUMABILITY, COMPLEXITY, THICKNESS, DENSITY, STICKINESS) appear once for every object (41 annotations each, 9.67 % apiece). CAPACITY appears 39 times (9.20 %), CONTENTS 38 (8.96 %), and SEALING 19 (4.48 %).

Category distribution (overall). Medium 36 (8.49 %), Non-sticky 36 (8.49 %), Contains 33 (7.78 %), Simple 31 (7.31 %), Non-containable 24 (5.66 %), Consumable 24 (5.66 %), Monochromatic 23

(5.42 %), High-density 21 (4.95 %), Low-density 20 (4.72 %), Hard 20 (4.72 %), Multicolored 18 (4.25 %), Light 18 (4.25 %), Soft 17 (4.01 %), Non-consumable 17 (4.01 %), Containable 15 (3.54 %), Sealed 13 (3.07 %), Thick 10 (2.36 %), Multi-object 10 (2.36 %), Heavy 10 (2.36 %), Thin 8 (1.89 %), Unsealed 6 (1.42 %), Empty 5 (1.18 %), Brittle 4 (0.94 %), Sticky 3 (0.71 %), Variable 2 (0.47 %).

Category distribution per property. CAPACITY: Non-containable 24 (61.5 %), Containable 15 (38.5 %). COLOR: Monochromatic 23 (56.1 %), Multicolored 18 (43.9 %). COMPLEXITY: Simple 31 (75.6 %), Multi-object 10 (24.4 %). CONSUMABILITY: Consumable 24 (58.5 %), Non-consumable 17 (41.5 %). CONTENTS: Contains 33 (86.8 %), Empty 5 (13.2 %). DENSITY: High-density 21 (51.2 %), Low-density 20 (48.8 %). HARDNESS: Hard 20 (48.8 %), Soft 17 (41.5 %), Brittle 4 (9.8 %). SEALING: Sealed 13 (68.4 %), Unsealed 6 (31.6 %). STICKINESS: Non-sticky 36 (87.8 %), Sticky 3 (7.3 %), Variable 2 (4.9 %). THICKNESS: Medium 23 (56.1 %), Thick 10 (24.4 %), Thin 8 (19.5 %). WEIGHT: Light 18 (43.9 %), Medium 13 (31.7 %), Heavy 10 (24.4 %).

Descriptor distribution (overall). Solid 44 (5.05 %); Lightweight 38 (4.36 %); Balanced 36, Smooth 36, Slippery 36 (4.13 % each); Filled 33, Occupied 33 (3.78 % each); Single-unit 31, Monolithic 31, Dense 31 (3.56 % each); Edible 24, Burnable 24, Disposable 24, Unperforated 24 (2.75 % each); Single Color 23, Neutral 23, Standard Thickness 23 (2.64 % each); Compact 21 (2.41 %); Buoyant 20, Bulky 20, Rigid 20 (2.29 % each); Featherweight 18, Gradient 18, Striped 18 (2.06 % each); Plush 17, Flexible 17, Reusable 17, Permanent 17 (1.95 % each); Hollow 15, Enclosable 15 (1.72 % each); Moderate 13, Airtight 13, Watertight 13 (1.49 % each); Sturdy 10, Assembled 10, Interconnected 10 (1.15 % each); Slim 8, Minimal Thickness 8 (0.92 % each); Open 6, Can-leak 6 (0.69 % each); Vacant 5, Void 5 (0.57 % each); Fragile 4, Breakable 4 (0.46 % each); Adhesive 3, Tacky 3 (0.34 % each); Temporary Stickiness 2, Conditional Adhesion 2 (0.23 % each).

Descriptor distribution per property & category. The synthetic generator enforces symmetric pairings: every category co-occurs with exactly two descriptors that split its count evenly—for instance, Containable objects are half *Hollow* and half *Enclosable*; Consumable items distribute equally among *Edible*, *Burnable*, and *Disposable*; Brittle objects are evenly *Fragile* and *Breakable*; analogous 50 % pairings hold across all remaining property–category combinations.

OpenImages

The *OpenImages* split aggregates 10 506 property annotations covering 679 everyday-object images for each of the 12 properties, i.e. 8 148 image–property pairs in total. Annotator effort is uneven but broad: Annot.,7 contributed 2 037 labels (19.4 %), Annot.,4 — 1 928 (18.4 %), Annot.,11 — 1 821 (17.3 %), Annot.,1 and 9 — 1 358 each (12.9 % ea.), Annot.,10 — 694 (6.6 %), Annot.,5 — 585 (5.6 %), Annot.,3 — 319 (3.0 %), Annot.,8 — 214 (2.0 %), Annot.,6 — 192 (1.8 %).

CAPACITY. All 679 images carry a CAPACITY label: Containable 321 (47.28 %), Non-containable 317 (46.69 %), Don't-know 35 (5.15 %), Not-applicable 6 (0.88 %). Descriptors cluster in two symmetrical pairs—*Hollow*/*Enclosable* (321 each, 25.16 % apiece) and *Solid*/*Unperforated* (317 each, 24.84 %).

COLOR. 818 colour judgements (often double-annotated) span the same image set. Categories: Multicolored 300 (36.67 %), Metallic 260 (31.78 %), Monochromatic 188 (22.98 %), Matte 59 (7.21 %), Don't-know 10 (1.22 %), Not-applicable 1. Descriptors: *Gradient* and *Striped* 300 each (18.59 %), *Glossy* and *Shiny* 260 each (16.11 %), *Single Color* 188 (11.65 %).

COMPLEXITY. 1 140 annotations—Multi-object 883 (77.46 %), Simple 242 (21.23 %), Don't-know 10, Invalid-format 5. Descriptors: *Assembled* / *Interconnected* 883 each (39.24 %), *Single-unit* / *Monolithic* 242 each (10.76 %).

CONSUMABILITY. Every image is labelled once: Non-consumable 633 (93.23 %), Consumable 41 (6.04 %), Invalid-format 4, Not-applicable 1. Descriptors split into reusable pairs—*Reusable*/*Permanent* 633 each (45.57 %) versus *Edible*/*Burnable*/*Disposable* 41 each (2.95 %).

CONTENTS. 679 labels: Contains 249 (36.67 %), Empty 149 (21.94 %), Not-applicable 149 (21.94 %), Don't-know 130 (19.15 %), Invalid-format 2. Descriptors: *Filled*/*Occupied* 249 each (31.28 %); *Vacant*/*Void* 149 each (18.72 %).

DENSITY. High-density 412 (60.68 %), Low-density 248 (36.52 %), Not-applicable 12, Don't-know 6, Variable 1. Descriptors mirror the split—*Dense*/*Compact* 412 each (31.16 %) versus *Lightweight*/*Buoyant* 248 each (18.76 %); one image is uniquely *Adjustable*.

HARDNESS. Hard 297 (43.74 %), Brittle 160 (23.56 %), Don't-know 126 (18.56 %), Soft 86 (12.67 %), Not-applicable 10. Descriptor pairs: *Solid*/*Rigid* 297 each (27.35 %), *Fragile*/*Breakable* 160 each (14.73 %), *Plush* 86 (7.92 %).

ORIENTATION. Vertical 496 (55.92 %), Horizontal 241 (27.17 %), Multi-directional 70 (7.89 %) plus 70 identical Invalid-format rows, Don't-know 8, Not-applicable 2. Descriptors: *Upright*/*Standing* 496 each (30.73 %), *Flat*/*Reclined* 241 each (14.93 %), *Rotational* 70 (4.34 %).

SEALING. Unsealed 495 (56.83 %), Sealed 351 (40.30 %), Don't-know 16 (1.84 %), Not-applicable 5 (0.57 %), Invalid-format 4 (0.46 %). Descriptors partition cleanly: *Open*/*Can leak* 495 each (29.26 %), *Airtight*/*Watertight* 351 each (20.74 %).

STICKINESS. 1 358 labels (two annotators × all images): Non-sticky 1 097 (80.78 %), Sticky 244 (17.97 %), Don't-know 15, Variable 2. Descriptors: *Smooth*/*Slippery* 1 097 each (40.84 %), *Adhesive*/*Tacky* 244 each (9.08 %), *Temporary Stickiness* 2 (0.07 %).

THICKNESS. Thick 258 (38.00 %), Medium 220 (32.40 %), Thin 163 (24.01 %), Not-applicable 27 (3.98 %), Don't-know 11 (1.62 %). Descriptors: *Sturdy*/*Bulky* 258 each (20.12 %), *Standard Thickness*/*Balanced* 220 each (17.16 %), *Slim* 163 (12.71 %).

WEIGHT. Heavy 482 (35.49 %), Light 443 (32.62 %), Medium 426 (31.37 %), Not-applicable 3, Don't-know 2, Dynamic 2. Descriptors: *Bulky*/*Dense* 482 each (17.81 %), *Featherweight*/*Lightweight* 443 each (16.37 %), *Moderate* 426 (15.74 %).

C.2 Affordance

OpenImages

Across 116 objects every image is annotated once, giving 116 affordance rows produced by seven annotators. Most images list three affordances (61 entries, 52.6 %), 50 list two (43.1 %), four list one (3.5 %) and one lists none. The ten most frequent affordances are: *Hold* 36 (12.5 %), *Holding* 11 (3.8 %), *Open/Close* 9 (3.1 %), *Cook* 9 (3.1 %), *Turn on/off* 8 (2.8 %), *Hold items* 6 (2.1 %), *Pour* 6 (2.1 %), *Fill* 5 (1.7 %), *Manipulating controls* 4 (1.4 %) and *Hold food* 4 (1.4 %).

Real Robot

Sixty-eight scenario pairs each have one affordance row, totalling 68 sets. Half of the scenarios list three affordances (34, 50 %), 29 list two (42.7 %) and five list one (7.4 %). Across all 170 recorded affordance slots the most common actions are: *act as weight* 29 (17.6 %), *Contain things* 12 (7.3 %), *scrape things* 10 (6.1 %), *stick things* 7 (4.2 %), *add thickness* 7 (4.2 %), *act as cushion* 7 (4.2 %), followed by fifteen further affordances occurring five or six times each. Slot-wise patterns highlight typical triplets such as *Contain things / act as cushion / act as weight* (7 cases, 10.3 %), and frequent pairs like *stick things / add thickness* or *break things / act as weight*. Slot 3 is often left blank (34 empty entries, 50 %).

Robocasa

The synthetic set covers 41 household objects. Eight objects list a single affordance (19.5 %), eighteen list two (43.9 %) and fifteen list three (36.6 %), giving 89 affordance mentions overall. *edible* dominates slot 1 (23 occurrences, 56.1 %) and is the single most frequent affordance overall (24, 27 %). Other common actions are *cookable* 10 (11.2 %), *garnish* and *can be used to stir things* 4

each (4.5 %), *can contain things*, *can be used to pour things*, *stackable* and *can be contain things* 3 each (3.4 %). All remaining 26 affordances appear once or twice ($\leq$2.5 % each), illustrating the long-tail synthetically injected diversity. The most common triplet is *edible / cookable / ∅* (10 objects, 24.4 %), followed by *edible / ∅ / ∅* (8, 19.5 %).

C.3 Constraints

Real Robo

This constraint Dataset contains 53 question–answer pairs, one per scenario. The nine distinct questions appear with the following frequencies: “Can we keep the ball inside the penstand?” 13 (24.53 %); “Can we keep the pen inside the penstand?” 10 (18.87 %); “Can you keep the food on the plate?” 8 (15.09 %); “Can you reverse the stacking of the objects?” 8 (15.09 %); “Can you write on the notepad using the marker?” 6 (11.32 %); “Can the robot stack the object near the right hand on the object near the left hand?” 4 (7.55 %); “Can the robot stack the object near the left hand on the object near the right hand?” 2 (3.77 %); “Can the robot stack the object away from it on the object near it?” 1 (1.89 %); and the lower-case duplicate “can you keep the food on the plate?” 1 (1.89 %).

All responses are negative and distributed across nineteen phrasings: “No the cube won’t balance on the pyramid.” 14 (26.42 %); “No the penstand is inverted.” 11 (20.75 %); “No the the penstand is inverted.” 4 (7.55 %); “No the penstand is not upright.” 4 (7.55 %); “No the box is on the plate.” 2 (3.77 %); “No the the penstand is not upright.” 2 (3.77 %); “No the plate is inverted.” 2 (3.77 %); “No the marker is closed.” 2 (3.77 %); “No the notepad is inside the cup.” 2 (3.77 %); plus nine single-occurrence answers covering cube–pyramid balance, covered openings, inverted or closed objects, and misplaced items.

Keyword extraction highlights the chief obstacles: “penstand” 23 mentions, “inverted” 20, and the instability trio “cube/balance/pyramid” 15 each, followed by “box” 9, “plate” 8, “closed” 7, “upright” 6, and sporadic references to notepad, cup, marker, inside, covered openings, under-placement and table contact.

Mapping these words to constraint types shows that inverted-orientation issues account for 20 cases (37.74 %); balance on a pyramid for 15 (28.30 %); object closure for 7 (13.21 %); non-upright alignment for 6 (11.32 %); containment failures (“inside”, “covered”, “under”, “on table”) and other special cases each represent $\leq$4 % of the set. Overall, tasks are blocked chiefly because penstands or plates are upside-down, cubes cannot balance on pyramids, or target objects are sealed or mis-aligned.

Mujoco

The Mujoco constraint Dataset contains 4 sub domains wach with 3 camera views. For each view we sample 10 different scenes configurations.

D Additional Model Evaluation Results

D.1 Prompt Design

Notations like "(T)" or "CoT" in the result tables (e.g., for Claude 3.7 Sonnet (T), o4-mini-high (T)) indicate the application of a Chain-of-Thought prompting strategy, where models were explicitly instructed to "think step by step" or provide reasoning before their final answer. The syntax for prompts are shown in Section B.2 B.3 B.4

D.2 Properties Evaluations

Beyond direct querying, we investigated the influence of prompting strategies, specifically Chain-of-Thought (CoT), on the performance of VLMs in understanding object properties. Table 8 presents the accuracies for various models when employing CoT prompting, which can be compared against their direct query performance shown in Table 7 (our main property results table with new data).

Chain-of-Thought Efficacy: A Mixed Bag for Property Recognition. Our analysis reveals that the impact of CoT prompting on property recognition is model-dependent and not uniformly beneficial across all properties or models. For instance, ‘Claude 3.7 Sonnet‘ shows a notable improvement with

Table 6: Properties accuracy (%) of leading VLMs across twelve distinct object property categories

Model	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
Claude 3.5 Sonnet	17.8	0.0	0.4	0.3	31.9	0.0	42.3	15.8	2.7	0.0	52.0	0.0
Claude 3.7 Sonnet	88.1	20.2	34.0	91.4	23.5	36.7	37.0	48.7	66.4	96.6	59.2	32.6
Claude 3.7 Sonnet (T)	81.3	6.7	38.4	93.8	22.3	9.0	23.4	24.0	50.9	73.8	46.2	15.0
Gemini 2.0 Flash 001	59.4	19.7	84.8	7.0	35.3	58.0	43.9	57.6	56.1	38.2	24.3	40.8
Gemini 2.5 Flash P	54.9	26.9	47.3	11.0	28.8	40.1	31.1	41.1	58.9	74.5	29.2	27.1
Gemini 2.5 Pro P**	48.9	27.0	47.4	23.7	34.1	43.2	16.7	33.1	57.2	23.2	32.6	31.2
Llama 3.2 90B Vision I	35.6	13.1	33.3	1.3	14.8	25.0	12.8	47.5	30.2	23.1	26.8	4.2
Llama 4 Maverick	53.0	36.2	52.5	69.6	34.9	47.0	14.6	53.9	90.0	93.6	37.9	37.6
Llama 4 Scout	43.3	30.4	12.6	0.2	0.6	51.1	18.6	31.7	84.9	9.5	28.3	36.4
GPT-4.1 Mini	70.1	26.6	85.0	59.9	28.4	43.2	18.1	45.6	64.0	91.9	52.3	24.1
GPT-4.1	10.9	13.8	38.1	5.3	29.0	25.9	27.8	42.3	91.0	35.3	37.0	4.4
o4-mini-high (T)	1.2	17.1	62.7	15.6	0.2	26.4	26.2	35.2	72.7	60.6	23.6	4.7
Qwen VL Plus	50.0	25.0	66.7	0.0	0.0	50.0	0.0	0.0	50.0	0.0	0.0	66.7
Qwen2.5 VL	53.2	21.9	34.2	9.0	20.7	9.6	42.3	57.1	61.8	70.7	66.6	18.7

Table 7: Properties Accuracy for Humanoid dataset

Model	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
Claude 3.7 Sonnet	74.6	47.8	47.3	93.0	30.3	55.7	55.7	59.7	13.2	79.1	39.3	48.3
Claude 3.5 Sonnet	83.6	50.2	48.8	89.6	28.9	52.7	55.2	58.7	19.4	83.6	42.8	50.7
Gemini 2.0 Flash 001	76.6	55.2	49.3	63.2	39.8	46.8	54.7	41.3	38.2	66.7	53.2	40.3
Gemini 2.5 Flash P	71.6	53.2	56.2	74.1	27.9	40.3	63.2	65.2	37.5	41.8	42.3	33.8
GPT-4.1*	76.1	51.2	52.7	66.7	55.7	58.2	64.2	60.7	43.8	81.6	41.8	43.3
GPT-4.1 Mini	55.2	36.3	47.3	75.1	36.3	40.3	60.2	58.7	15.3	49.3	38.8	26.9
Llama 4 Maverick	82.1	43.8	46.3	82.6	77.1	57.7	54.2	47.8	40.3	62.7	40.8	59.2
Llama 4 Scout	81.6	51.2	45.8	62.2	60.2	43.3	51.2	54.2	36.1	73.6	44.3	37.8
Llama 3.2 90B VI*	59.7	37.3	36.3	39.3	51.2	44.8	37.3	39.8	27.1	56.7	16.9	31.3
Qwen2.5 VL	31.3	47.8	46.3	27.9	22.9	4.5	35.8	34.3	2.8	5.0	15.4	24.9
Qwen VL Plus*	25.4	15.4	28.4	22.9	20.4	39.3	34.3	29.4	31.3	12.4	14.4	5.5
Grok 2 Vision	69.7	49.3	45.8	53.7	82.6	40.3	56.7	55.2	11.1	78.6	37.3	31.8
Grok Vision Beta*	7.5	4.5	4.5	8.0	1.5	5.0	4.5	3.0	1.4	7.0	3.5	1.0

Table 8: Properties accuracy using **chain-of-thought (COT) prompting**. (**) Subset of properties evaluated.

Model	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
Claude 3.5 Sonnet	21.5	2.6	4.4	4.2	35.2	1.2	43.2	20.0	6.4	3.5	52.2	4.9
Claude 3.7 Sonnet	89.9	21.3	36.8	94.6	24.6	41.5	41.7	50.2	69.8	100.0	63.3	35.1
Claude 3.7 Sonnet (T)	84.5	9.9	41.0	94.2	25.9	10.9	28.2	24.8	54.4	78.3	48.0	16.6
Gemini 2.0 Flash 001	62.5	20.6	86.0	7.8	36.5	62.5	48.5	60.3	57.9	39.9	24.9	44.8
Gemini 2.5 Flash P	57.6	30.3	47.8	15.5	30.4	43.2	32.9	45.8	60.3	76.4	31.8	28.4
Gemini 2.5 Pro P**	–	28.9	–	–	–	20.7	–	–	–	–	35.0	–
Llama 3.2 90B Vision I	36.1	17.8	37.8	4.4	18.5	27.6	12.9	51.0	34.3	23.5	27.9	6.4
Llama 4 Maverick	57.2	36.8	57.0	69.7	38.1	49.9	19.2	56.7	94.1	96.5	41.0	38.0
Llama 4 Scout	44.9	32.1	15.7	0.7	4.8	53.6	20.7	32.2	89.2	9.6	28.7	36.5
GPT-4.1 Mini	71.0	27.1	86.7	63.2	31.6	44.1	22.9	48.7	68.6	94.2	52.5	28.9
GPT-4.1	11.1	18.4	39.8	5.4	31.8	29.5	32.7	47.2	93.5	38.0	39.6	8.6
o4-mini-high	4.1	21.5	66.3	16.4	0.6	27.8	30.5	35.9	75.2	62.0	26.3	8.1
Qwen VL Plus	53.4	26.3	71.2	2.7	1.8	50.1	1.5	2.6	54.0	3.0	3.2	68.7
Qwen2.5 VL	53.3	25.2	38.6	12.5	22.0	12.8	44.2	61.4	65.1	70.9	70.2	19.5

CoT on ‘Sealing (P9)‘ (from 13.2% direct to 69.8% CoT) and ‘Stickiness (P10)‘ (from 79.1% direct to 100.0% CoT). However, for the same model, CoT appears to slightly decrease performance on

'Density (P6)' (from 55.7% direct to 41.5% CoT). Its '(T)' variant in Table 8 (which is its CoT run) also shows improvements in some areas like 'Complexity (P3)'.

D.3 Affordance Evaluations

Table 9: Affordance Accuracy (%) of VLMs on recognizing **at least one correct affordance** for objects grouped by primary categories (Single-Category Mapping)

Model	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	H1	H2	H3
Claude 3.5 Sonnet	0.0	0.0	0.0	0.0	0.0	0.0	16.7	25.0	0.0	66.7	13.3	40.0	9.1	0.0	0.0	0.0	44.4	0.0	2.9	47.1	14.7
Claude 3.7 Sonnet (T)	0.0	5.6	0.0	30.0	0.0	0.0	0.0	0.0	11.1	0.0	6.7	20.0	18.2	0.0	0.0	0.0	0.0	0.0	2.9	54.4	10.3
Claude 3.7 Sonnet	100.0	0.0	0.0	20.0	0.0	0.0	0.0	0.0	11.1	66.7	0.0	20.0	22.7	100.0	0.0	0.0	33.3	0.0	2.9	58.8	11.8
Gemini 2.0 Flash 001	0.0	0.0	0.0	40.0	0.0	0.0	16.7	0.0	0.0	66.7	0.0	40.0	13.6	0.0	0.0	0.0	11.1	0.0	54.4	66.2	64.7
Gemini 2.5 Flash P	0.0	5.6	0.0	20.0	0.0	50.0	0.0	0.0	11.1	66.7	13.3	40.0	18.2	0.0	50.0	0.0	22.2	0.0	52.9	55.9	57.4
Gemini 2.5 Pro P	0.0	16.7	66.7	30.0	0.0	0.0	33.3	25.0	22.2	66.7	26.7	60.0	31.8	0.0	0.0	33.3	11.1	0.0	0.0	0.0	0.0
Llama 3.2 11B Vision I	100.0	22.2	0.0	30.0	0.0	50.0	33.3	0.0	22.2	66.7	0.0	0.0	13.6	0.0	50.0	33.3	33.3	0.0	20.5	27.9	25.0
Llama 3.2 90B Vision I	100.0	11.1	33.3	10.0	0.0	50.0	50.0	25.0	22.2	66.7	26.7	60.0	9.1	0.0	0.0	0.0	22.2	0.0	22.1	44.1	0.0
Llama 4 Maverick	0.0	22.2	33.3	50.0	0.0	100.0	50.0	0.0	33.3	66.7	26.7	100.0	31.8	0.0	0.0	33.3	11.1	100.0	20.6	39.7	23.5
Llama 4 Scout	0.0	11.1	66.7	50.0	0.0	50.0	50.0	25.0	33.3	66.7	53.3	60.0	54.6	100.0	50.0	0.0	33.3	0.0	20.6	27.9	26.5
GPT 4.1 Mini	0.0	5.6	0.0	30.0	0.0	0.0	50.0	25.0	0.0	100.0	13.3	60.0	36.4	0.0	0.0	0.0	55.6	0.0	20.6	57.4	25.0
GPT 4.1	0.0	5.6	0.0	20.0	0.0	0.0	16.7	25.0	0.0	0.0	6.7	60.0	18.2	0.0	0.0	0.0	33.3	0.0	48.5	67.6	45.6
o4-mini-high (T)	0.0	16.7	0.0	20.0	0.0	0.0	16.7	25.0	11.1	33.3	33.3	20.0	22.7	0.0	0.0	0.0	11.1	0.0	16.2	45.6	35.3
Qwen 2.5 VL	0.0	0.0	0.0	30.0	0.0	0.0	33.3	0.0	0.0	100.0	6.7	80.0	9.1	0.0	0.0	0.0	11.1	0.0	14.7	48.5	20.6
Qwen 3	0.0	5.5	0.0	30.0	0.0	0.0	33.3	25.0	0.0	100.0	0.0	60.0	13.6	0.0	0.0	0.0	44.4	0.0	4.4	1.4	8.8
Grok 2 Vision	0.0	5.6	33.3	50.0	0.0	0.0	0.0	0.0	11.1	100.0	6.7	20.0	13.6	100.0	50.0	0.0	0.0	0.0	44.1	47.1	41.2
Grok 2 Beta	0.0	5.6	0.0	10.0	0.0	0.0	0.0	0.0	11.1	0.0	13.3	20.0	4.6	0.0	0.0	0.0	33.3	100.0	8.8	8.8	7.4

Table 10: Accuracy (%) of VLMs on recognizing **all correct affordances** for objects grouped by primary categories (Single-Category Mapping) in PAC Bench. Categories C1-C18 are as defined in Table 3

Model	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18
Claude 3.5 Sonnet	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Claude 3.7 Sonnet (T)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Claude 3.7 Sonnet	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Gemini 2.0 Flash 001	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Gemini 2.5 Flash P	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Gemini 2.5 Pro P	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6.7	0.0	0.0	0.0	0.0	0.0	0.0
Llama 3.2 11B Vision I	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Llama 3.2 90B Vision I	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Llama 4 Maverick	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Llama 4 Scout	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.5	0.0	0.0	0.0	0.0	0.0
GPT 4.1 Mini	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
GPT 4.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	20.0	0.0	0.0	0.0	0.0	0.0	0.0
o4-mini-high (T)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Qwen VP	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Qwen 2.5 VL	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	11.1	0.0
Grok 2 Vision	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Grok 2 Beta	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Following Table 12 shows Accuracy (%) of VLMs on recognizing **atleast one affordances** for objects using **Single-Category Mapping** in PAC Bench. For the object classes 'Adhesive tape', 'Backpack', 'Band-aid', 'Bathroom accessory', 'Bathroom cabinet', 'Bathtub', 'Blender', 'Book', 'Bookcase', 'Bottle', 'Bowl', 'Box', 'Cabinetry', 'Can opener', 'Cart', 'Chair', 'Chest of drawers', 'Closet', 'Clothing', 'Coffeemaker', 'Container', 'Cooking spray', 'Countertop', 'Cupboard', 'Cutting board', 'Desk', 'Diaper', 'Dishwasher', 'Door', 'Door handle', 'Drawer', 'Drill (Tool)', 'Egg (Food)', 'Filing cabinet', 'Flashlight', 'Flowerpot', 'Food processor', 'Fork', 'Frying pan', 'Furniture', 'Gas stove', 'Glove', 'Grinder', 'Hammer', 'Home appliance', 'Infant bed', 'Jug', 'Kettle', 'Kitchen & dining room table', 'Kitchen appliance', 'Kitchen knife', 'Kitchen utensil', 'Knife', 'Ladder', 'Ladle', 'Laptop', 'Lavender (Plant)', 'Light bulb', 'Light switch', 'Measuring cup', 'Microwave oven', 'Milk', 'Mirror', 'Mixer', 'Mixing bowl', 'Mobile phone', 'Mug', 'Organ (Musical Instrument)', 'Oven', 'Paper towel', 'Pen', 'Pitcher (Container)', 'Plant', 'Plastic bag', 'Plate', 'Plumbing fixture', 'Power plugs and sockets', 'Pressure cooker', 'Refrigerator', 'Remote control', 'Scissors', 'Screwdriver', 'Serving tray', 'Shelf', 'Shower', 'Sink', 'Slow cooker', 'Soap dispenser', 'Spatula', 'Spice rack', 'Spoon', 'Stairs', 'Stool', 'Table', 'Tablet computer', 'Tableware', 'Tap', 'Toaster', 'Toilet', 'Toilet paper', 'Tool', 'Toothbrush', 'Torch', 'Towel', 'Toy', 'Waffle iron', 'Wardrobe', 'Washing machine', 'Waste container', 'Whisk', 'Window blind', 'Wok', 'Wood-burning stove', 'Wrench', 'Zucchini'.

Table 11: Accuracy (%) of VLMs on recognizing **at least one correct affordance** for objects using **Multi-Category Mapping** in PAC Bench.

Model	A1	A2	A3	A4	A5	C6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18
Claude 3.5 Sonnet	0.0	0.0	0.0	0.0	0.0	0.0	18.2	20.0	0.0	50.0	18.8	14.3	5.3	0.0	14.3	0.0	35.7	0.0
Claude 3.7 Sonnet (T)	0.0	5.6	0.0	27.3	0.0	0.0	0.0	0.0	9.1	0.0	12.5	14.3	10.5	0.0	0.0	12.5	7.1	0.0
Claude 3.7 Sonnet	100.0	0.0	0.0	18.2	0.0	0.0	0.0	0.0	9.1	50.0	6.2	14.3	13.2	100.0	14.3	12.5	35.7	0.0
Gemini 2.0 Flash 001	0.0	0.0	0.0	36.4	0.0	0.0	9.1	0.0	0.0	50.0	0.0	21.4	7.9	0.0	14.3	25.0	7.1	0.0
Gemini 2.5 Flash P	0.0	5.6	0.0	27.3	0.0	33.3	9.1	0.0	9.1	50.0	12.5	21.4	13.2	0.0	28.6	12.5	14.3	0.0
Gemini 2.5 Pro P	0.0	16.7	66.7	36.4	0.0	33.3	36.4	20.0	27.3	50.0	31.2	57.1	26.3	0.0	14.3	37.5	28.6	0.0
Llama 3.2 11B VI	100.0	22.2	0.0	27.3	0.0	33.3	18.2	20.0	18.2	50.0	0.0	7.1	18.4	0.0	42.9	50.0	28.6	0.0
Llama 3.2 90B VI	100.0	11.1	33.3	18.2	0.0	33.3	45.5	20.0	18.2	50.0	31.2	42.9	10.5	0.0	28.6	25.0	14.3	0.0
Llama 4 Maverick	0.0	22.2	33.3	54.5	0.0	66.7	36.4	0.0	36.4	50.0	31.2	57.1	23.7	0.0	28.6	62.5	21.4	100.0
Llama 4 Scout	0.0	11.1	66.7	54.5	0.0	33.3	54.5	40.0	27.3	50.0	56.2	35.7	36.8	100.0	42.9	50.0	42.9	0.0
GPT 4.1 Mini	0.0	5.6	0.0	36.4	0.0	0.0	27.3	20.0	0.0	75.0	12.5	42.9	23.7	0.0	28.6	25.0	50.0	0.0
GPT 4.1	0.0	5.6	0.0	27.3	0.0	0.0	9.1	20.0	0.0	0.0	12.5	35.7	13.2	0.0	28.6	12.5	35.7	0.0
o4-mini-high (T)	0.0	16.7	0.0	18.2	0.0	0.0	9.1	20.0	18.2	25.0	31.2	21.4	18.4	0.0	14.3	12.5	21.4	0.0
Qwen VP	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Qwen 2.5 VL	0.0	0.0	0.0	36.4	0.0	0.0	18.2	0.0	0.0	75.0	12.5	35.7	5.3	0.0	14.3	25.0	7.1	0.0
Grok 2 Vision	0.0	5.6	33.3	45.5	0.0	0.0	0.0	0.0	18.2	75.0	6.2	28.6	7.9	100.0	14.3	37.5	7.1	0.0
Grok Vision Beta	0.0	5.6	0.0	9.1	0.0	0.0	0.0	0.0	9.1	0.0	12.5	14.3	5.3	0.0	14.3	0.0	21.4	100.0

Table 12: Accuracy (%) of VLMs on recognizing **all correct affordances** for objects using **Multi-Category Mapping** in PAC Bench.

Model	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18
Claude 3.5 Sonnet	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Claude 3.7 Sonnet (T)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Claude 3.7 Sonnet	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Gemini 2.0 Flash 001	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Gemini 2.5 Flash P	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Gemini 2.5 Pro P	0.0	0.0	0.0	0.0	0.0	0.0	9.1	0.0	0.0	0.0	6.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Llama 3.2 11B VI	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Llama 3.2 90B VI	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Llama 4 Maverick	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Llama 4 Scout	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.6	0.0	0.0	0.0	0.0	0.0
GPT 4.1 Mini	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
GPT 4.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	7.1	0.0	0.0	0.0	0.0	0.0	0.0
o4-mini-high (T)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Qwen VP	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Qwen 2.5 VL	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	14.3	0.0	7.1	0.0
Grok 2 Vision	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Grok Vision Beta	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Object Class	Claude 3.5 S.	Claude 3.7 S. (T)	Claude 3.7 S.	Gemini 2.0 F001	Gemini 2.5 FP	Gemini 2.5 PP	Llama 3.2 11B (1)	Llama 3.2 11B (2)	Llama 3.2 90B VI	Llama 4 Mav.	Llama 4 Sct.	GPT 4.1 Mini	GPT 4.1	o4-mini-high (T)	Qwen VP	Qwen 2.5 VL	Grok 2 Vis.	Grok V. Beta
Adhesive Tape	0.0	0.0	100.0	0.0	0.0	0.0	0.0	100.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Backpack	0.0	0.0	0.0	0.0	0.0	100.0	100.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0
Band-Aid	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Bathroom Accessory	0.0	0.0	0.0	0.0	100.0	0.0	100.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	100.0
Bathroom Cabinet	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Bathtub	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Blender	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Book	0.0	0.0	0.0	0.0	0.0	0.0	100.0	100.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Bookcase	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Bottle	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	100.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Bowl	0.0	0.0	0.0	0.0	100.0	0.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	0.0	0.0	100.0	100.0	0.0
Box	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	100.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0
Cabinetry	0.0	0.0	0.0	0.0	100.0	100.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Can Opener	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Cart	0.0	0.0	0.0	0.0	0.0	100.0	100.0	0.0	100.0	100.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Chair	0.0	100.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0
Chest Of Drawers	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0
Closet	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	100.0	100.0	100.0	0.0	100.0	0.0	0.0	0.0	0.0

(continued on next page)

(continued from previous page)

Object Class	Claude 3.5 S.	Claude 3.7 S. (T)	Claude 3.7 S.	Gemini 2.0 F001	Gemini 2.5 FP	Gemini 2.5 PP	Llama 3.2 11B (1)	Llama 3.2 11B (2)	Llama 3.2 90B VI	Llama 4 Mav.	Llama 4 Sct.	GPT 4.1 Mini	GPT 4.1	o4-mini-high (T)	Qwen VP	Qwen 2.5 VL	Grok 2 Vis.	Grok V. Beta
Clothing	0.0	0.0	0.0	0.0	0.0	0.0	100.0	100.0	100.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Coffeemaker	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Container	100.0	0.0	0.0	100.0	0.0	100.0	0.0	0.0	100.0	0.0	0.0	100.0	100.0	0.0	0.0	0.0	0.0	0.0
Cooking Spray	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Countertop	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Cupboard	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	100.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0
Cutting Board	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Desk	0.0	0.0	0.0	0.0	0.0	100.0	100.0	0.0	100.0	100.0	100.0	100.0	100.0	0.0	0.0	100.0	0.0	100.0
Diaper	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Dishwasher	0.0	0.0	0.0	0.0	0.0	100.0	100.0	0.0	0.0	100.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0
Door	100.0	100.0	100.0	0.0	0.0	100.0	100.0	0.0	100.0	100.0	100.0	0.0	100.0	0.0	0.0	100.0	0.0	0.0
Door Handle	100.0	0.0	0.0	100.0	100.0	100.0	0.0	0.0	100.0	100.0	100.0	100.0	0.0	0.0	0.0	100.0	100.0	0.0
Drawer	100.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	100.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Drill (Tool)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Egg (Food)	100.0	0.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	0.0	0.0	100.0	0.0	0.0	0.0	100.0	100.0	0.0
Filing Cabinet	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Flashlight	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Flowerpot	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Food Processor	0.0	0.0	0.0	0.0	0.0	0.0	100.0	100.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Fork	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Frying Pan	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	100.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Furniture	100.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	100.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0
Gas Stove	0.0	0.0	0.0	0.0	0.0	0.0	100.0	100.0	0.0	0.0	100.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0
Glove	0.0	0.0	0.0	0.0	100.0	0.0	100.0	0.0	0.0	100.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Grinder	0.0	0.0	0.0	0.0	100.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0
Hammer	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Home Appliance	0.0	100.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Infant Bed	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Jug	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0
Kettle	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Kitchen	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0
Kitchen Appliance	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Kitchen Knife	0.0	100.0	100.0	100.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	100.0	100.0	100.0	0.0	0.0	0.0	0.0
Kitchen Utensil	0.0	100.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	100.0
Knife	100.0	0.0	100.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0
Ladder	100.0	0.0	100.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	100.0	0.0	0.0
Ladle	0.0	0.0	100.0	100.0	0.0	100.0	0.0	0.0	0.0	0.0	100.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0
Laptop	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Lavender (Plant)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0
Light Bulb	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	100.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Light Switch	0.0	0.0	0.0	0.0	0.0	100.0	100.0	0.0	0.0	0.0	100.0	0.0	0.0	100.0	0.0	0.0	100.0	100.0
Measuring Cup	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Microwave Oven	0.0	0.0	0.0	0.0	100.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0
Milk	100.0	0.0	0.0	0.0	100.0	0.0	100.0	100.0	100.0	100.0	100.0	100.0	0.0	0.0	0.0	100.0	100.0	0.0
Mirror	0.0	0.0	0.0	0.0	0.0	100.0	100.0	0.0	100.0	0.0	0.0	100.0	100.0	0.0	0.0	0.0	0.0	0.0
Mixer	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Mixing Bowl	100.0	0.0	100.0	0.0	0.0	0.0	100.0	100.0	0.0	0.0	100.0	100.0	0.0	100.0	0.0	0.0	100.0	0.0
Mobile Phone	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Mug	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Organ (Musical Instrument)	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0
Oven	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Paper Towel	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Pen	0.0	0.0	0.0	0.0	100.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0
Pitcher (Container)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	100.0	100.0	100.0	100.0	0.0	0.0	0.0	100.0	0.0	0.0
Plant	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Plastic Bag	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Plate	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	100.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Plumbing Fixture	0.0	0.0	0.0	0.0	100.0	100.0	100.0	0.0	100.0	100.0	100.0	100.0	100.0	0.0	0.0	100.0	0.0	0.0
Power Plugs And Sockets	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0
Pressure Cooker	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0
Refrigerator	0.0	0.0	0.0	0.0	0.0	0.0	100.0	100.0	100.0	100.0	100.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0
Remote Control	0.0	100.0	0.0	0.0	0.0	0.0	0.0	100.0	100.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Scissors	100.0	0.0	0.0	100.0	100.0	0.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	0.0	0.0	0.0	0.0
Screwdriver	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	100.0	100.0	0.0	0.0	0.0	0.0	0.0
Serving Tray	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Shelf	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Shower	0.0	0.0	100.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Sink	0.0	100.0	0.0	100.0	100.0	100.0	100.0	0.0	0.0	100.0	0.0	100.0	0.0	0.0	0.0	100.0	100.0	0.0
Slow Cooker	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

(continued on next page)

(continued from previous page)

Object Class	Claude 3.5 S.	Claude 3.7 S. (T)	Claude 3.7 S.	Gemini 2.0 F001	Gemini 2.5 FP	Gemini 2.5 PP	Llama 3.2 11B (1)	Llama 3.2 11B (2)	Llama 3.2 90B VI	Llama 4 Mav.	Llama 4 Sct.	GPT 4.1 Mini	GPT 4.1	o4-mini-high (T)	Qwen VP	Qwen 2.5 VL	Grok 2 Vis.	Grok V. Beta
Soap Dispenser	0.0	0.0	0.0	100.0	0.0	0.0	0.0	100.0	0.0	100.0	100.0	100.0	0.0	0.0	0.0	100.0	100.0	0.0
Spatula	100.0	100.0	100.0	0.0	0.0	0.0	100.0	0.0	0.0	100.0	100.0	100.0	100.0	100.0	0.0	0.0	100.0	0.0
Spice Rack	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0
Spoon	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Stairs	0.0	0.0	0.0	100.0	0.0	0.0	100.0	0.0	0.0	100.0	0.0	100.0	100.0	100.0	0.0	100.0	0.0	100.0
Stool	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0
Table	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	100.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0
Tablet Computer	0.0	0.0	100.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Tableware	0.0	100.0	0.0	0.0	100.0	100.0	100.0	0.0	0.0	100.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Tap	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0
Toaster	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Toilet	0.0	100.0	0.0	100.0	0.0	0.0	100.0	0.0	0.0	100.0	100.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0
Toilet Paper	0.0	0.0	0.0	0.0	0.0	100.0	0.0	100.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0
Tool	0.0	0.0	100.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0
Toothbrush	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Torch	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Towel	0.0	100.0	100.0	100.0	0.0	100.0	100.0	0.0	100.0	100.0	100.0	100.0	100.0	100.0	0.0	100.0	100.0	0.0
Toy	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0
Waffle Iron	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Wardrobe	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Washing Machine	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	100.0	0.0	0.0	100.0	0.0
Waste Container	0.0	0.0	0.0	0.0	0.0	100.0	100.0	100.0	100.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0
Whisk	0.0	0.0	0.0	0.0	0.0	100.0	0.0	100.0	0.0	100.0	100.0	100.0	100.0	100.0	0.0	0.0	0.0	0.0
Window Blind	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Wok	0.0	0.0	0.0	100.0	0.0	100.0	0.0	0.0	0.0	0.0	100.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0
Wood-Burning Stove	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Wrench	0.0	0.0	0.0	0.0	0.0	100.0	100.0	0.0	100.0	0.0	100.0	100.0	100.0	0.0	0.0	0.0	0.0	100.0
Zucchini	0.0	0.0	100.0	100.0	0.0	100.0	100.0	0.0	0.0	100.0	100.0	100.0	0.0	100.0	0.0	100.0	100.0	0.0

Following Table 14 shows Accuracy (%) of VLMs on recognizing **all correct affordances** for objects using **Single-Category Mapping** in PAC Bench.

Object Class	Claude 3.5 S.	Claude 3.7 S. (T)	Claude 3.7 S.	Gemini 2.0 F001	Gemini 2.5 FP	Gemini 2.5 PP	Llama 3.2 11B (1)	Llama 3.2 11B (2)	Llama 3.2 90B VI	Llama 4 Mav.	Llama 4 Sct.	GPT 4.1 Mini	GPT 4.1	o4-mini-high (T)	Qwen VP	Qwen 2.5 VL	Grok 2 Vis.	Grok V. Beta
Adhesive Tape	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Backpack	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Band-Aid	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Bathroom Accessory	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Bathroom Cabinet	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Bathtub	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Blender	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Book	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Bookcase	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Bottle	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Bowl	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Box	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Cabinetry	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Can Opener	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Cart	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Chair	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Chest Of Drawers	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Closet	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Clothing	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Coffeemaker	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Container	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Cooking Spray	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Countertop	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

(continued on next page)

(continued from previous page)

Object Class	Claude 3.5 S.	Claude 3.7 S. (T)	Claude 3.7 S.	Gemini 2.0 F001	Gemini 2.5 FP	Gemini 2.5 PP	Llama 3.2 11B (1)	Llama 3.2 11B (2)	Llama 3.2 90B VI	Llama 4 Mav.	Llama 4 Sct.	GPT 4.1 Mini	GPT 4.1	o4-mini-high (T)	Qwen VP	Qwen 2.5 VL	Grok 2 Vis.	Grok V. Beta
Cupboard	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Cutting Board	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Desk	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Diaper	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Dishwasher	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Door	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Door Handle	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Drawer	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Drill (Tool)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Egg (Food)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Filing Cabinet	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Flashlight	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Flowerpot	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Food Processor	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Fork	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Frying Pan	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Furniture	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Gas Stove	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Glove	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Grinder	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Hammer	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Home Appliance	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Infant Bed	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Jug	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Kettle	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Kitchen	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Kitchen Appliance	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Kitchen Knife	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Kitchen Utensil	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Knife	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Ladder	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Ladle	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Laptop	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Lavender (Plant)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Light Bulb	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Light Switch	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Measuring Cup	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Microwave Oven	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Milk	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Mirror	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Mixer	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Mixing Bowl	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Mobile Phone	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Mug	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Organ (Musical Instrument)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Oven	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Paper Towel	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Pen	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Pitcher (Container)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Plant	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Plastic Bag	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Plate	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Plumbing Fixture	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Power Plugs And Sockets	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Pressure Cooker	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Refrigerator	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Remote Control	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Scissors	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0
Screwdriver	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Serving Tray	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Shelf	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Shower	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Sink	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Slow Cooker	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Soap Dispenser	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Spatula	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Spice Rack	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Spoon	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Stairs	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0

(continued on next page)

(continued from previous page)

Object Class	Claude 3.5 S.	Claude 3.7 S. (T)	Claude 3.7 S.	Gemini 2.0 F001	Gemini 2.5 FP	Gemini 2.5 PP	Llama 3.2 11B (1)	Llama 3.2 11B (2)	Llama 3.2 90B VI	Llama 4 Mav.	Llama 4 Sct.	GPT 4.1 Mini	GPT 4.1	o4-mini-high (T)	Qwen VP	Qwen 2.5 VL	Grok 2 Vis.	Grok V. Beta
Stool	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Table	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Tablet Computer	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Tableware	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Tap	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Toaster	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Toilet	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Toilet Paper	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Tool	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Toothbrush	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Torch	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Towel	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Toy	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Waffle Iron	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Wardrobe	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Washing Machine	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Waste Container	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Whisk	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Window Blind	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Wok	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Wood-Burning Stove	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Wrench	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Zucchini	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

D.4 Constraint Evaluations

Table 15: Examples of Real-World Humanoid Constraint Scenarios from PAC Bench. Each scenario includes a question posed about a potential action and the ground-truth constraint explanation. Scenarios are captured using synchronized Agent View (from robot's perspective) and Side View cameras.

Views Provided	Question Posed	Ground-Truth Constraint Explanation
Agent View (cam_0) Side View (cam_1)	Can the robot stack the object near the right hand on the object near the left hand?	No the cube won't balance on the pyramid.
Agent View (cam_0) Side View (cam_1)	Can we keep the ball inside the penstand?	No the the penstand is inverted.
Agent View (cam_0) Side View (cam_1)	Can we keep the ball inside the penstand?	No the the opening of the penstand is covered by the hand.
Agent View (cam_0) Side View (cam_1)	Can you keep the food on the plate?	No the box is closed.
Agent View (cam_0) Side View (cam_1)	Can you write on the notepad using the marker?	No the marker is closed.
Agent View (cam_0) Side View (cam_1)	Can you keep the food on the plate?	No the box is on the plate.

E Human Survey

This section describes how we gathered and filtered the human–annotated labels that accompany our three image collections: (i) a single–image subset of OpenImages, (ii) the *Real-Robo* dual-view humanoid dataset, and (iii) the *RoboCasa* synthetic renders. Across all datasets we collected categorical judgements for **15 physical-property ontologies** (e.g. *Weight*, *Hardness*, *Capacity*) together with free-form affordances and, where relevant, environment constraints. The same label

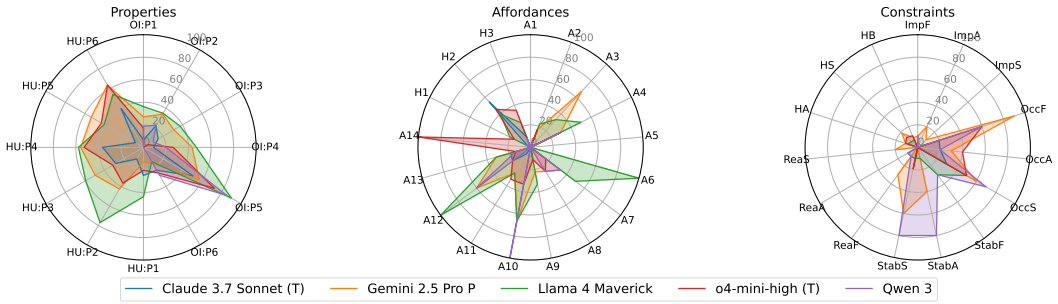

Figure 10: Performance of best models from each family

set, category order, and keyboard shortcuts were used everywhere to ensure a uniform annotation experience (see Figures 11–15).

E.1 Annotation Pipelines

Single-image (OpenImages). We created one Label Studio[4] project per property. Each task presents a pre-cropped object (bounding box supplied) and radio-button choices covering the ontology plus *Don't Apply* and *Don't Know*. Annotators select exactly one option that best reflects the object's *current visual state* (e.g. a sauce-coated spoon is marked *Sticky*); an example interface is shown in Figure 15. Hot-keys (1–4 to pick a category, `Ctrl/Cmd+\Enter` to advance) support rapid, fatigue-free labelling. The per-property job dashboard is illustrated in Figure 14. Open-vocabulary affordances could not be captured with fixed radio buttons, so they were instead filled into a shared Google Sheet ($\leq$3 verbs per image ID).

Dual-view (*Real-Robo* & *RoboCasa*). Label Studio does not support paired views, so we developed a lightweight Python/Tkinter GUI that shows the left/right camera frames side-by-side (Figures 12 and 13). The GUI mirrors the exact ontologies, category ordering, and hot-keys of the single-image pipeline and appends three affordance text boxes plus a drop-down for task-level constraints. For completeness, the corresponding single-image TkInter variant used for synthetic objects is depicted in Figure 11.

E.2 Annotation Schedule and Effort

Each property job comprises $\sim$680 items and takes $\approx$40 minutes per annotator after a brief tutorial. All properties were labelled by at least two annotators to enable later consensus filtering (see below); several critical properties were triple-annotated when calendar time allowed. The total annotation effort is roughly $15 properties \times 2.2 annotators \times 40 min \approx 22$ person-hours for OpenImages and 7 person-hours for the dual-view collections.

E.3 Quality Control

We employ a strict *unanimity filter*: for every image (or view-pair) the final label is retained only if *all* assigned annotators agreed. Disagreements are discarded from the main release (and provided as a separate "disagreement split") to guarantee that the benchmark set reflects high-confidence, noise-free supervision.

E.4 Annotators

All personal identifiers are withheld to preserve double-blind review integrity.

[4] `https://labelstud.io`

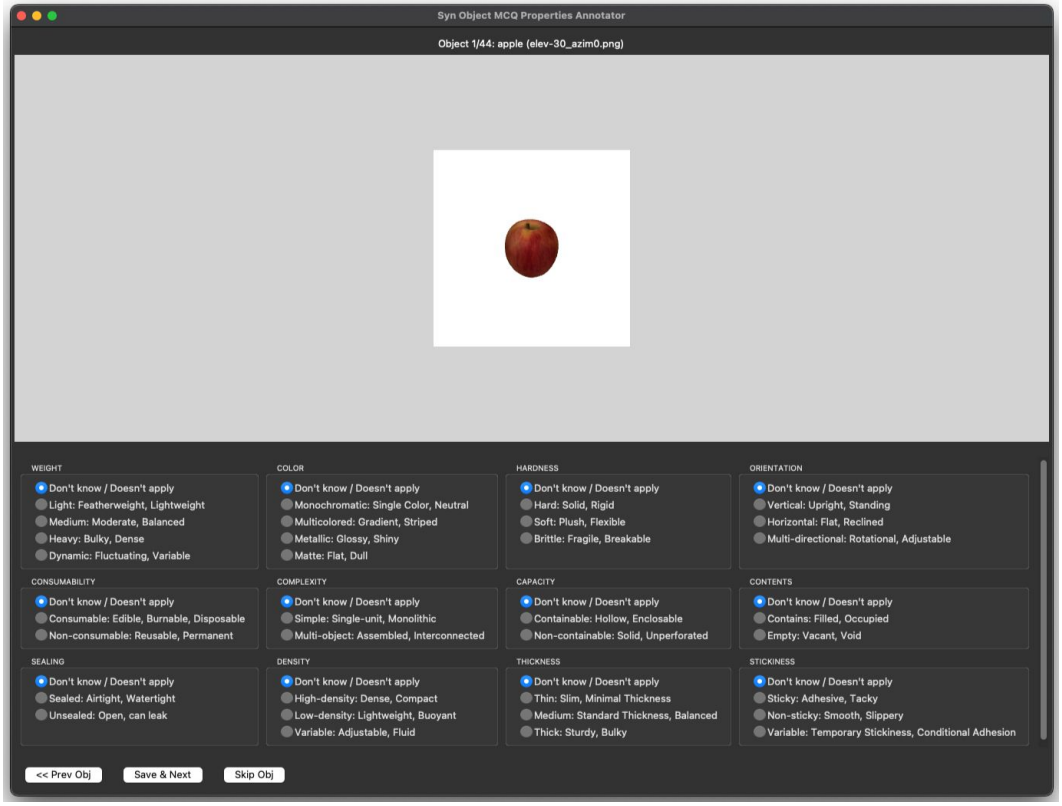

Figure 11: TkInter single-image property annotator (synthetic objects).

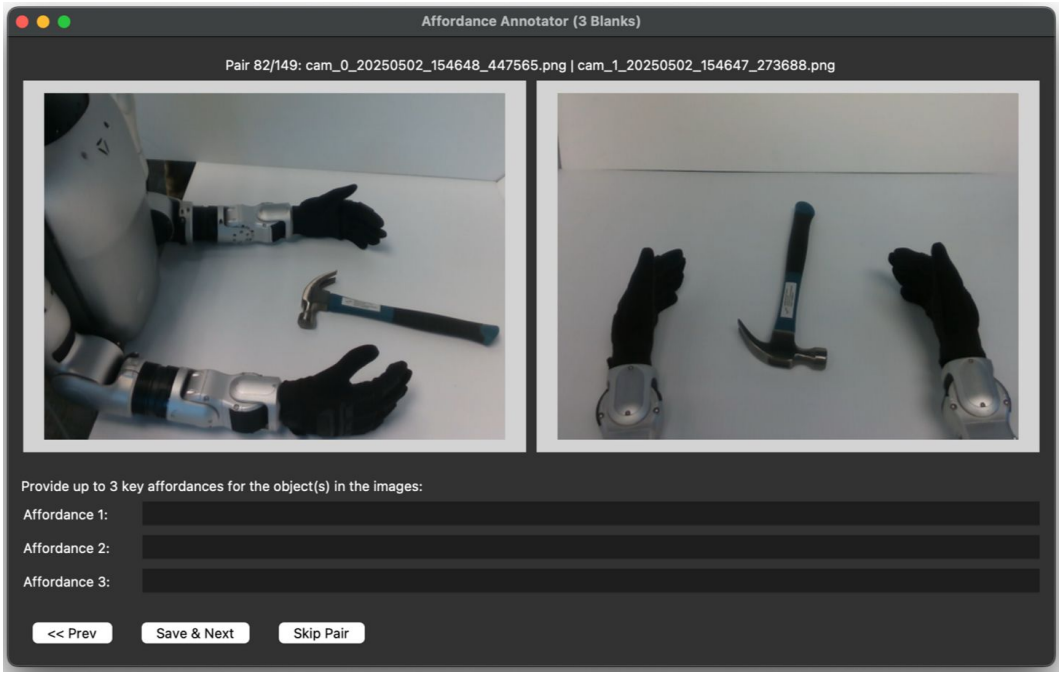

Figure 12: TkInter dual-view affordance annotator.

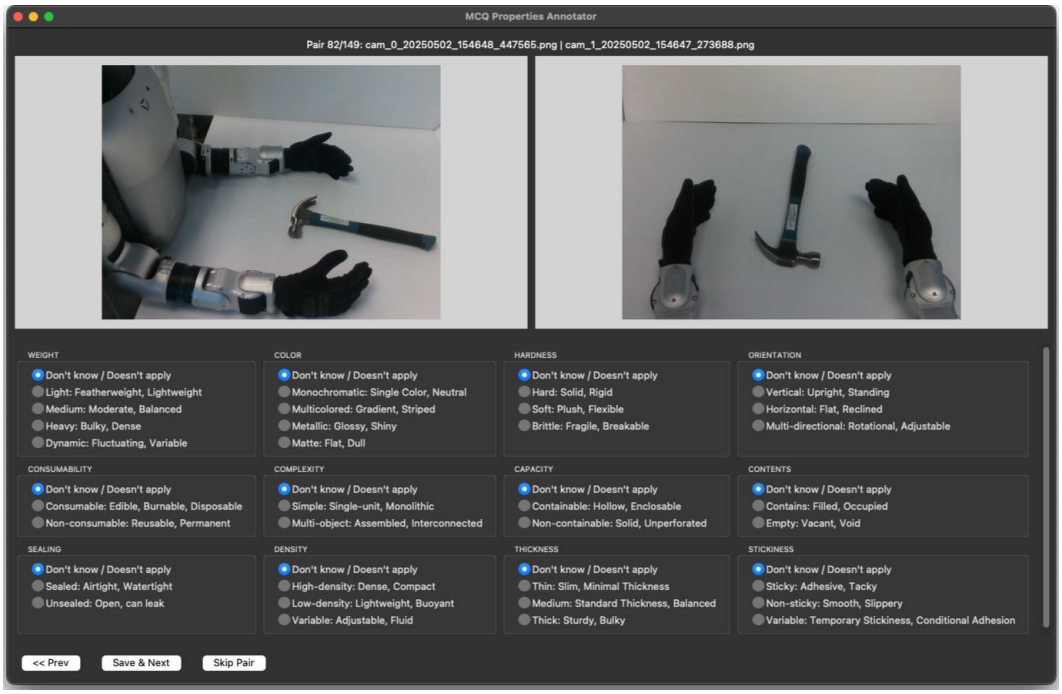

Figure 13: TkInter dual-view property annotator (Real-Robo / RoboCasa).

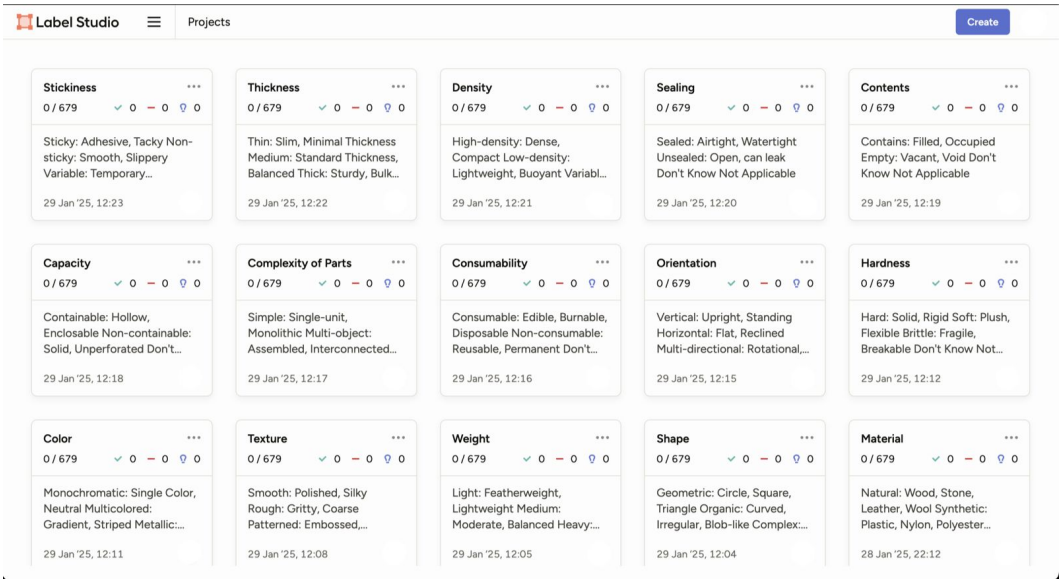

Figure 14: Label Studio project dashboard with 15 property jobs.

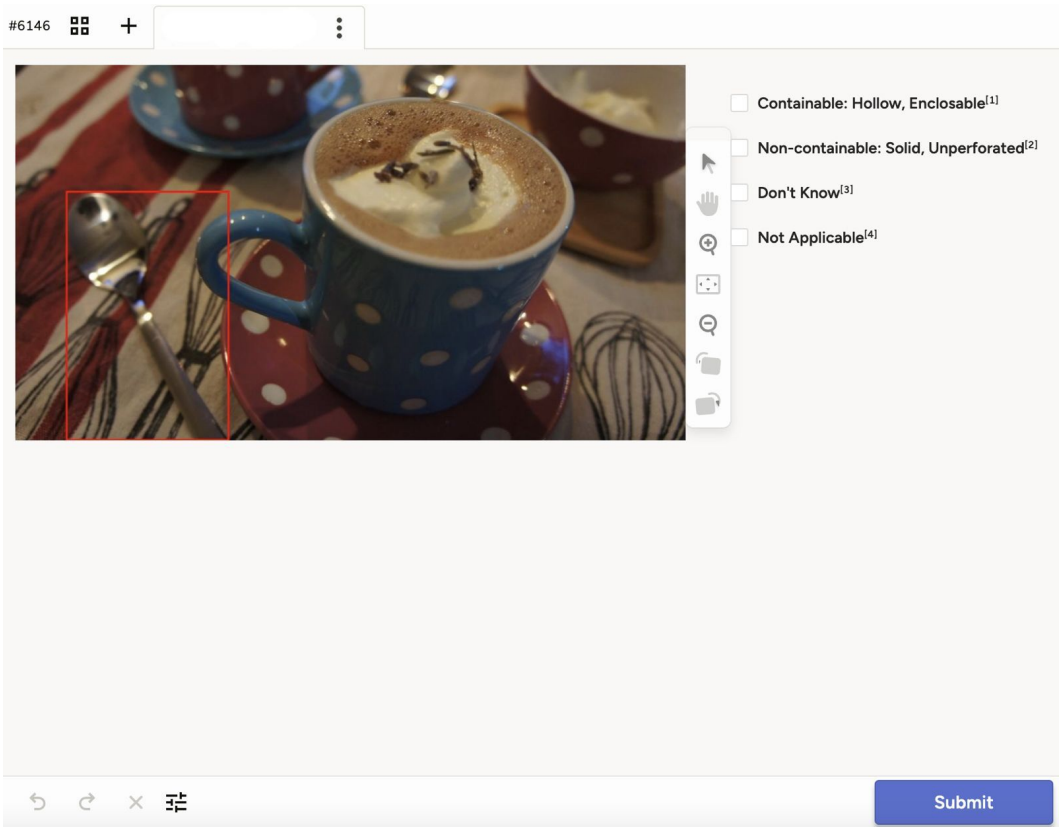

Figure 15: Label Studio image view with bounding box and radio-button options.