A Significance, Scope, and Clarifications

A.1 Why CreativityBench Matters

A Missing Dimension in Current Evaluation. Recent progress in LLMs has been measured primarily through analytical reasoning, tool execution, and long-horizon planning. Our work focuses on a different and still under-measured capability: **creative tool use grounded in physical affordances**. The central question is not whether a model can produce a plausible plan in words, but whether it can identify **which object** and **which physical property** make a non-obvious solution actually usable. We view this as an important and practical dimension of intelligence, especially for agents that must operate under constraints, limited resources, or unforeseen situations.

From Object Selection to Mechanism-Level Reasoning. A key contribution of CreativityBench is that it moves evaluation beyond coarse object-level plausibility. Existing settings often reward selecting a generally relevant object. In contrast, our benchmark requires models to localize the relevant **part**, connect it to **attributes**, and recover the **affordance mechanism** that enables success. This finer level of grounding is exactly where current models show their largest performance drop, which suggests that the benchmark is exposing a real and previously under-tested weakness rather than merely repackaging existing commonsense tasks.

A Structured Resource for Studying Grounded Creativity. The benchmark is supported by a structured affordance knowledge base linking entities, parts, attributes, and affordances. This organization is important for two reasons. First, it enables systematic task construction at scale while preserving an interpretable latent solution path. Second, it makes failure analysis much more informative: when a model fails, we can ask whether it selected the wrong entity, the wrong part, the wrong attribute basis, or an implausible use condition. This is a major advantage over open-ended creativity evaluations that are difficult to diagnose or reproduce.

Implications for Future Models and Agents. We believe the significance of this benchmark extends beyond one dataset. For LLMs, it highlights the limits of strong general reasoning when physical grounding and unconventional repurposing are required. For embodied systems, it points to a core challenge in robust real-world problem solving: useful behavior often depends on recognizing latent affordances under constraints, not just following canonical object functions. More broadly, the benchmark offers a concrete testbed for studying how grounded knowledge, comparative search, and flexible recombination interact in creative problem solving.

A.2 Clarifications of Concerns

Scope and Methodological Role of LLM-Assisted Construction. In our setting, LLM assistance is used primarily as a **scaling mechanism for structured data construction**, not as a substitute for the benchmark definition itself. The benchmark is anchored in an explicit ontology including entity, part, attribute, affordance, and conditions, and the resulting tasks are generated from this structured representation rather than from unconstrained free-form synthesis. This design reduces arbitrariness: the LLM does not define creativity in an open-ended way, but instantiates a predefined schema that makes the benchmark more systematic, auditable, and extensible.

Why the Benchmark Still Measures More Than Retrieval. We agree that physical commonsense is a necessary ingredient, but the benchmark requires more than retrieving a known object-use association. Each task is built so that the model must infer a **non-obvious** but physically grounded connection among a goal, a candidate part, relevant attributes, and conditions of use. In other words, the model must move from surface familiarity to **mechanism-sensitive repurposing**. This is precisely why performance is much lower on

the full gold selection than on coarse entity selection. The benchmark therefore targets a narrower and more operational notion of creativity: discovering a usable, constraint-compatible, non-canonical function from latent affordance structure.

Single-Gold Structure as a Deliberate Evaluation Choice. Creative real-world problems may admit multiple plausible solutions. Therefore, our benchmark does not simply assume the uniqueness of the sampled gold affordance as the solution for the benchmark task. Instead, our construction pipeline explicitly includes **gold verification** through intra-entity and inter-entity comparison, with candidate tasks rejected or filtered when an alternative is judged preferable. The purpose of this step is not to claim that every real-world problem has exactly one valid solution, but to construct a controlled evaluation set where one solution path is sufficiently well-supported to enable rigorous comparison across models. In this sense, the single-gold structure is a methodological choice for **measurement clarity**, not a denial of the open-ended nature of creativity.

Interpreting Human Performance Carefully. The human results may appear surprising at first glance, since humans do not outperform the best model on the strict gold metric. We do not interpret this as evidence that the benchmark is misaligned with human reasoning. Rather, it reflects an important property of the current evaluation format: humans receive compact textual descriptions of entities, parts, and attributes instead of direct perceptual access to objects. This makes the task more like **symbolic affordance inspection** than everyday embodied improvisation. Importantly, this does not weaken the benchmark's diagnostic value. Instead, it shows that the benchmark isolates a specific reasoning challenge: identifying grounded affordances from structured descriptions, a task that remains cognitively demanding for humans in the absence of perceptual shortcuts. That makes the setting a meaningful stress test rather than an easy commonsense exercise.

Domain Coverage and the Value of a Focused Household Setting. We view our benchmark's focused scope is a strength at the current stage. Household environments provide rich, diverse, and widely understandable object ecosystems in which creative repurposing naturally arises. By restricting the domain, we can control entity distributions, affordance granularity, and distractor sampling far more carefully than would be possible in a highly heterogeneous open-domain benchmark. This improves internal validity and makes failure patterns easier to interpret. We therefore present CreativityBench not as the final word on all forms of creativity, but as a principled and scalable benchmark for one important, grounded subtype of creative problem solving.

Cross-Model Comparisons Should Be Read Structurally, Not Just Numerically. Differences across model families can be sensitive to prompting and implementation choices, as is true of most benchmark comparisons. Our central conclusions, however, do not rely on a brittle ordering between two closely matched systems. Instead, the paper highlights several structural patterns that remain consistent across models: a substantial drop from entity-level to part-level accuracy, weaker physical grounding than action plausibility, pronounced sensitivity to affordance commonality, and limited improvement from standard inference-time interventions. These recurring trends are more informative than any single leaderboard position. Accordingly, the benchmark's main contribution is not simply to show that one model outperforms another, but to uncover a shared failure mode that appears across both proprietary and open-source model families.

Final Take Away. CreativityBench is intentionally scoped, structurally grounded, and diagnostically designed. It does not aim to solve every ambiguity in evaluating creativity, nor does it claim that creative intelligence can be reduced to one benchmark. Our contribution is more precise: we isolate an important form of creative tool use, formalizes it through affordance-based structure, and shows that current models remain substantially weaker at this capability than their general reasoning progress might suggest. In our view, this combination of conceptual focus, structured construction, and consistent empirical findings is exactly what makes the benchmark useful.

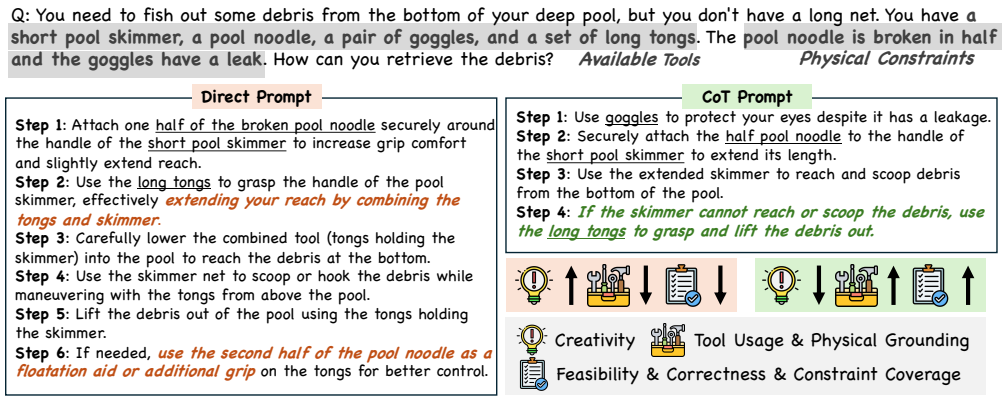


Figure 11: Qualitative comparison between generated responses from direct and CoT prompts.

B Preliminary Experiments Details

B.1 Experiment setting Details

Our preliminary experiment is designed as a controlled comparison between two prompting strategies on 100 creative tool-use tasks sampled from MacGyver (Tian et al., 2024). In the **direct prompt** setting, the model is asked to propose a feasible solution under the task constraints without any prescribed intermediate reasoning steps, thereby testing its implicit ability to connect task requirements with tool functions. In the **structured affordance-level CoT** setting, the model is instead guided through an explicit reasoning pipeline that includes listing the available tools, decomposing each tool into parts, inferring relevant physical properties, deriving possible affordances, justifying each action step, and validating the final solution against the stated constraints.

We evaluate outputs using six criteria: **Correctness**, **Feasibility**, **Physical Grounding**, **Constraint Coverage**, **Tool Usage**, and **Creativity**, under both absolute scoring (1–5 per dimension) and pairwise relative comparison between the two prompting strategies. Please see the prompts below for more details. We use GPT-4.1-mini as the target model and GPT-5.2 as the judge model, employing temperature 0.0 to guarantee deterministic outputs. All of other settings we follow the original MacGyver paper's protocols.

We also provide a case study to compare the direct generated response and CoT response. Please refer to Figure 11.

B.2 Core Prompt Details

Directive Prompt for Creative Tool-Use

```
You are an expert at creative physical tool-use reasoning.
Given the task description below, produce a feasible solution.

Rules:
1. Use only tools/items explicitly available in the task description.
2. Respect physical constraints if the task description restricts physical attributes such as size or
state.
3. Invent new tools using tools/items explicitly available in the task description if it is needed.
4. Provide practical steps that can actually be executed.
5. Do not involve any unnecessary steps to achieve the task's goal.
6. If a complete solution is impossible, return the best partial plan and explain why it cannot be
completed.

TASK DESCRIPTION:
{problem}
```

```
Return JSON:
{{
  "solvable": "Yes or No",
  "solvable_explanation": "1-3 sentences about why the given task is solvable or not",
  "solution_steps": ["Step 1: ...", "Step 2: ...", ...],
  "final_solution": "One concise paragraph that summarizes the full approach.",
  "used_tools": ["tool 1", "tool 2", ...],
  "constraint_handling": [{{"constraint": "...", "handling": "..."}}, ...]
}}
```

CoT Prompt for Creative Tool-Use

```
You are an expert at creative physical tool-use reasoning.
Solve the task by explicitly reasoning over tool parts and affordances under constraints.

Rules:
1. Use only tools/items explicitly available in the task description.
2. Respect physical constraints if the task description restricts physical attributes such as size or
state.
3. Invent new tools using tools/items explicitly available in the task description if it is needed.
4. Provide practical steps that can actually be executed.
5. Do not involve any unnecessary steps to achieve the task's goal.
6. If a complete solution is impossible, return the best partial plan and explain why it cannot be
completed.

Required reasoning procedure:
1. State the task goal and concrete success condition.
2. List all available tools/items from the prompt (no additions).
3. For each relevant tool, identify the key part(s), infer physical properties, and derive part-level
affordances useful for this task.
4. Build a step-by-step plan where each step references tool parts and the affordance being used.
5. Validate each step against stated constraints (e.g., broken/unusable items, size mismatch, blocked
function, state limitations).
6. Keep the plan practical and minimal with no unnecessary actions.

TASK DESCRIPTION:
{problem}

Return JSON:
{{
  "task_goal": "...",
  "success_condition": "...",
  "identified_constraints": ["...", "..."],
  "tool_inventory": [
    {{
      "tool": "...",
      "relevant_parts": [
        {{
          "part": "... or NA",
          "inferred_physical_properties": ["...", "..."],
          "affordances_for_task": ["...", "...", ...],
          "usable_under_constraints": "Yes or No",
        }},
        ...
      ]
    }},
    ...
  ],
  "reasoning_plan": [
    {{
      "step": 1,
      "action": "...",
      "tool_parts_used": ["tool:part", "..."],
      "affordance_used": ["...", "..."],
      "explanation": "1 sentence about why it works"
    }},
    {{
      "step": 2,
      "action": "...",
      "tool_parts_used": ["tool:part", "..."],
      "affordance_used": ["...", "..."],
      "explanation": "1 sentence about why it works"
    }},
    ...
  ],
  "solvable": "Yes or No",
```

```
  "solvable_explanation": "1-3 sentences about why the given task is solvable or not",
  "solution_steps": ["Step 1: ...", "Step 2: ...", ...],
  "final_solution": "One concise paragraph that summarizes the full approach.",
  "used_tools": ["tool 1", "tool 2", ...],
  "constraint_handling": [{{"constraint": "...", "handling": "..."}}, ...]
  "creative_reasoning_summary": "1-3 sentences about novelty and practicality."
}}
```

Prompt for LLM-as-Judge Absolute Evaluation: Correctness

```
You are an expert evaluator for creative physical tool-use reasoning quality.
Your task is to evaluate the CORRECTNESS of a candidate solution against a ground-truth solution.

**DEFINITION**: Correctness measures whether the solution actually solves the stated task goal. Mark it as
 correct if the solution reaches the task goal even if it is different from ground-truth solution.

**SCORING RUBRIC (1-5)**:
- 1 = Completely wrong. The solution does not address the task at all, solves a different problem, or
produces an outcome opposite to what is needed.
    - Example: Task asks to "open a locked door using available tools" and the candidate describes
painting the door.
- 2 = In between 1 and 3
- 3 = Partially correct. The solution moves toward the goal but is incomplete, solves only a sub-part of
the task, or requires unstated assumptions to work. The core objective is addressed but not fully achieved
.
    - Example: Task asks to "move a heavy boulder off a path" and the candidate correctly identifies using
 a lever but never specifies where to place the fulcrum, leaving the solution unresolved.
- 4 = In between 3 and 4
- 5 = Fully correct. The solution completely and unambiguously achieves the task goal. All steps are
logically consistent and lead to the desired outcome.
    - Example: Task asks to "lift a car to change a tire without a jack" and the candidate correctly uses
a sturdy plank wedged under the frame with a rock as fulcrum, applies body weight, and secures the car
before swapping the tire.

**TASK DESCRIPTION**:
{problem}

**GROUND-TRUTH SOLUTION**:
{ground_truth_solution}

**CANDIDATE SOLUTION**:
{candidate_solution}

**CANDIDATE SOLUTION USED TOOLS**:
{used_tools}

Return JSON:
{{
  "correct": "Yes or No",
  "correct_score": 1-5,
  "correct_rationale": "1-2 sentences for rationale."
}}
```

Prompt for LLM-as-Judge Absolute Evaluation: Feasibility

```
You are an expert evaluator for creative physical tool-use reasoning quality.
Your task is to evaluate the FEASIBILITY of a candidate solution against a ground-truth solution.

**DEFINITION**: Feasibility measures whether the solution can be physically executed under the stated
constraints.

**SCORING RUBRIC (1-5)**:
- 1 = Completely infeasible. The solution requires violating a stated constraint, ignores key physical
limits, or depends on impossible actions.
    - Example: Constraint says "no cutting tools available" and the candidate instructs cutting a rope in
half.
- 2 = In between 1 and 3
- 3 = Mostly feasible. The solution is executable in principle but contains one or two steps that stretch
physical plausibility or lightly brush against a constraint without outright violating it. Would work
under ideal or slightly relaxed conditions.
    - Example: TSolution requires balancing a long beam on a narrow pivot point -- physically possible but
 practically difficult without more stabilization.
- 4 = In between 3 and 4
```

```
- 5 = Fully feasible. Every step can be realistically performed by a person with ordinary strength and
skill using only the available tools, within all stated constraints. No step requires superhuman effort or
 ignores stated limitations.
    - Example: All actions use objects explicitly listed, forces are within human capability, and
constraints (e.g., "no power tools," "indoors only") are strictly respected throughout.

**TASK DESCRIPTION**:
{problem}

**GROUND-TRUTH SOLUTION**:
{ground_truth_solution}

**CANDIDATE SOLUTION**:
{candidate_solution}

**CANDIDATE SOLUTION USED TOOLS**:
{used_tools}

Return JSON:
{{
  "feasible": "Yes or No",
  "feasible_score": 1-5,
  "feasible_rationale": "1-2 sentences for rationale."
}}
```

Prompt for LLM-as-Judge Absolute Evaluation: Physical Grounding

```
You are an expert evaluator for creative physical tool-use reasoning quality.
Your task is to evaluate the PHYSICAL GROUNDING of a candidate solution against a ground-truth solution.

**DEFINITION**: Physical grounding measures whether the solution correctly leverages the physical
properties, geometry, and mechanics of the objects involved.

**SCORING RUBRIC (1-5)**:
- 1 = Physically nonsensical. The solution misuses objects in ways that contradict their basic physical
properties (e.g., using a soft cloth as a rigid lever, treating a hollow tube as load-bearing).
    - Example: "Use the inflatable pool toy as a wedge to hold the car axle in place" -- an inflatable toy
 cannot bear that load.
- 2 = In between 1 and 3
- 3 = Partially grounded. Most physical reasoning is sound, but one or two steps misidentify a relevant
property (e.g., ignoring friction, misjudging weight distribution, or using the wrong part of an object).
    - Example: Correctly identifies a metal rod as a lever but places the fulcrum at the rod's midpoint
when the task geometry requires it near one end for sufficient mechanical advantage.
- 4 = In between 3 and 4
- 5 = Fully grounded. The solution correctly identifies and exploits the relevant physical attributes of
every tool (material strength, shape, weight, surface texture, rigidity, etc.) and applies correct
mechanical principles (leverage, friction, torque, buoyancy, etc.) throughout.
    - Example: "Insert the flat-head screwdriver under the lid's lip -- the thin, rigid blade provides a
moment arm; pressing down on the handle rotates the blade and pries the lid upward with minimal force."

**TASK DESCRIPTION**:
{problem}

**GROUND-TRUTH SOLUTION**:
{ground_truth_solution}

**CANDIDATE SOLUTION**:
{candidate_solution}

**CANDIDATE SOLUTION USED TOOLS**:
{used_tools}

Return JSON:
{{
  "physical_grounding": "Yes or No",
  "physical_grounding_score": 1-5,
  "physical_grounding_rationale": "1-2 sentences citing specific object properties or mechanics addressed
or missed."
}}
```

Prompt for LLM-as-Judge Absolute Evaluation: Constraint Coverage

```
You are an expert evaluator for creative physical tool-use reasoning quality.
Your task is to evaluate the CONSTRAINT COVERAGE of a candidate solution against a ground-truth solution.

**DEFINITION**: Constraint coverage measures whether the solution explicitly handles every constraint
stated in the task.

**SCORING RUBRIC (1-5)**:
- 1 = Ignores constraints. One or more hard constraints are violated outright or never acknowledged.
    - Example: Task says "use only items found in the kitchen" and the candidate fetches a tool from a
garage.
- 2 = In between 1 and 3
- 3 = Partial coverage. Most constraints are respected but one is violated or only implicitly handled
without explanation. The solution would likely still work but shows incomplete awareness.
    - Example: Task says "complete the task silently and without electricity" -- the candidate avoids
electric tools but produces loud banging with no acknowledgment of the noise constraint.
- 4 = In between 3 and 4
- 5 = Full coverage. Every constraint is explicitly acknowledged and the solution demonstrates a clear,
deliberate strategy for satisfying each one. No constraint is left unaddressed or violated.
    - Example: "Since we cannot use water (constraint 1) and must work in under two minutes (constraint 2)
, I will use dry sand to smother the flame and apply it immediately using the nearby bucket."

**TASK DESCRIPTION**:
{problem}

**GROUND-TRUTH SOLUTION**:
{ground_truth_solution}

**CANDIDATE SOLUTION**:
{candidate_solution}

**CANDIDATE SOLUTION USED TOOLS**:
{used_tools}

Return JSON:
{{
  "constraint_coverage": "Yes or No",
  "constraint_coverage_score": 0-5,
  "constraint_coverage_rationale": "1-2 sentences identifying which constraints were met or violated."
}}
```

Prompt for LLM-as-Judge Absolute Evaluation: Tool Usage

```
You are an expert evaluator for creative physical tool-use reasoning quality.
Your task is to evaluate the TOOL USAGE of a candidate solution against a ground-truth solution.

**DEFINITION**: Tool usage measures whether the solution uses only the tools and objects explicitly
available in the task description, and whether each tool is used in a sensible, purposeful way. You can
also refer to CANDIDATE SOLUTION USED TOOLS to evaluate the score.

**SCORING RUBRIC (1-5)**:
- 1 = Fabricates tools. The solution relies on objects not listed in the task or invents tools out of thin
 air.
    - Example: Task lists "rope, wooden plank, and a brick" and candidate says "use the pulley system" --
no pulley was provided.
- 2 = In between 1 and 3
- 3 = Mostly within bounds with minor overreach. The solution uses available tools correctly for the most
part but includes one object that is not explicitly listed yet could plausibly be assumed present.
    - Example: "the ground" as a fulcrum when only portable objects were listed.
- 4 = In between 3 and 4
- 5 = Perfect tool fidelity. Every object used is explicitly listed in the task description. Each tool is
used for a clear, stated purpose and no phantom tools are introduced. The solution makes full, purposeful
use of what is available without waste or omission.
    - Example: Task provides "a broom, two bricks, and a length of string" and the candidate uses objects
presented in the task in a coherent plan that references each by name and function.

**TASK DESCRIPTION**:
{problem}

**GROUND-TRUTH SOLUTION**:
{ground_truth_solution}

**CANDIDATE SOLUTION**:
{candidate_solution}
```

```
**CANDIDATE SOLUTION USED TOOLS**:
{used_tools}

Return JSON:
{{
  "tool_usage": "Yes or No",
  "tool_usage_score": 1-5,
  "tool_usage_rationale": "1-2 sentences noting any phantom tools or unused available tools.",
}}
```

Prompt for LLM-as-Judge Absolute Evaluation: Creativity

```
You are an expert evaluator for creative physical tool-use reasoning quality.
Your task is to evaluate the CREATIVE REASONING of a candidate solution against a ground-truth solution.

**DEFINITION**: Creative reasoning measures how inventively the solution repurposes or combines available
tools beyond their obvious, default use.

**SCORING RUBRIC (1-5)**:
- 1 = Low Creativity. The solution follows an obvious, expected approach with no novel tool-use.
    - Example: "Use the hammer to drive the nail."
- 2 = In between 1 and 3
- 3 = Medium Creativity. The solution repurposes at least one object in a non-obvious way, or combines
tools in a moderately clever configuration that goes beyond the most straightforward reading of the task.
    - Example: Using a belt as an improvised tourniquet and winding peg.
- 4 = In between 3 and 4
- 5 = High Creativity. The solution demonstrates genuinely inventive thinking: unexpected combinations of
objects, multi-step mechanical chains, or insights that exploit subtle physical properties most people
would overlook. The approach is surprising yet clearly sound.
    - Example: Using a glass of water as a lens to focus sunlight onto tinder, combined with a foil
wrapper as a reflector to intensify the beam.

**TASK DESCRIPTION**:
{problem}

**GROUND-TRUTH SOLUTION**:
{ground_truth_solution}

**CANDIDATE SOLUTION**:
{candidate_solution}

**CANDIDATE SOLUTION USED TOOLS**:
{used_tools}

Return JSON:
{{
  "creative_reasoning": "Yes or No",
  "creativity_score": 1-5,
  "creativity_rationale": "1-2 sentences explaining what makes the approach creative or conventional."
}}
```

Prompt for LLM-as-Judge Relative Evaluation

```
You are an expert evaluator for creative physical tool-use reasoning quality.
Your task is to compare TWO candidate solutions for the SAME task, using the ground-truth solution as
reference.

IMPORTANT:
- You are NOT checking wording similarity. Judge functional quality and practical validity.
- A solution can be strong even if it differs from the ground-truth approach.
- Use the following criteria:
    - Correctness: Whether the plan actually solves the task objective.
    - Feasibility: Whether the plan can be physically executed under stated constraints.
    - Physical Grounding: Whether it uses realistic object properties/mechanics correctly.
    - Constraint Coverage: Whether it handles all explicit constraints.
    - Tool Usage: Whether it uses only available tools appropriately and purposefully.
    - Creative Reasoning: Novel, non-obvious but valid repurposing/combination of tools.
    - Overall: Holistic quality across all above dimensions.

For EACH criterion:
1. Determine winner:
   - "win" if solution1_score > solution2_score
   - "lose" if solution2_score > solution1_score
```

```
   - "tie" if both are equal
2. Give a short rationale (1-2 sentences).

TASK DESCRIPTION:
{problem}

GROUND-TRUTH SOLUTION:
{ground_truth_solution}

CANDIDATE SOLUTION1 (DEFAULT PROMPT OUTPUT):
{solution1}

CANDIDATE SOLUTION2 (COT PROMPT OUTPUT):
{solution2}

Return STRICT JSON:
{{
  "correctness": {{
    "winner": "win or lose or tie",
    "rationale": "1-2 sentences."
  }},
  "feasibility": {{
    "winner": "win or lose or tie",
    "rationale": "1-2 sentences."
  }},
  "physical_grounding": {{
    "winner": "win or lose or tie",
    "rationale": "1-2 sentences."
  }},
  "constraint_coverage": {{
    "winner": "win or lose or tie",
    "rationale": "1-2 sentences."
  }},
  "tool_usage": {{
    "winner": "win or lose or tie",
    "rationale": "1-2 sentences."
  }},
  "creativity": {{
    "winner": "win or lose or tie",
    "rationale": "1-2 sentences."
  }},
  "overall": {{
    "winner": "win or lose or tie",
    "rationale": "1-2 sentences."
  }},
  "short_summary": "2-4 sentences summarizing key tradeoffs between default and CoT."
}}
```

B.3 Textual v.s. Visual Grounding in Creative Tool Use

We showed that adding an affordance-decomposition CoT scaffold does not reliably improve creative reasoning in text-only tasks, suggesting the bottleneck is not "missing step-by-step procedure" but limited grounded affordance representations and compositional recombination (Section 3). Unlike text-only tasks, which explicitly specify the task goal, available tools, and physical constraints in symbolic form, we further examine whether the influence of affordance-level CoT persists under a visual+text setting. In this multimodal condition, the text provides only the task goal and non-visible constraints, while the image presents available tools and visible physical constraints. Since affordances are never given at inference time, the model must infer object attributes and derive affordances internally in both settings; however, the visual condition requires extracting observable properties (e.g., shape, relative size, geometry) from perceptual input rather than reading them directly from text.

To disentangle these factors, we run a 2×2 ablation over the same model (GPT-4.1-mini): input modality (Text-only vs. Vision+Text) × prompting strategy (Direct vs. Affordance-CoT). In the text-only setting, the description explicitly lists the goal, available tools, and constraints. In the vision+text setting, the text specifies only the goal and non-visible constraints (e.g., temperature), while the image conveys available objects and visible constraints. This design isolates whether performance gains come from (1) richer grounding signals, (2) reasoning scaffolds, or (3) their interaction.

Scenes	Entities	Parts	Physical Attributes	State Attributes	Affordances
Bathroom	483	3,264	35,859	15,551	19,584
Bedroom	468	3,447	37,896	16,311	20,682
Dining Room	477	3,099	34,044	14,581	18,594
Garage	474	3,393	37,254	16,349	20,357
Garden	477	3,087	33,909	14,881	18,522
Home Office	468	3,489	38,373	16,526	20,934
Kitchen	498	3,111	34,188	14,887	18,666
Living Room	471	3,348	36,795	15,886	20,088
Total	**3,816**	**26,238**	**288,318**	**124,972**	**157,427**

Table 5: Scene-wise statistics of the affordance knowledge base. Each scenario is annotated with entities, parts, physical attributes, state attributes, and affordances.

C Annotation Pipeline Details

C.1 Overview of Annotation Stages

Our annotation pipeline is a staged, LLM-assisted procedure that transforms a set of environment scenarios into richly annotated object-part datasets for creativity-oriented affordance analysis. The process proceeds from entity grounding to structural decomposition, then to physical and state characterization, and finally to part-level functional affordance annotation and full-entity assembly. Each stage consumes the output of the previous stage, so that downstream annotations are grounded in upstream structure and attributes rather than generated independently.

Stage A: Scenario-Grounded Entity Generation. We begin with a predefined set of eight everyday indoor scenarios including Kitchen, Living Room, Bedroom, Bathroom, Garage, Home Office, Dining Room, Garden. For each scenario, we sample a fixed number of **specific single objects** (not broad categories and not complex multi-object systems). To maintain diversity while reducing near-duplicates, we apply semantic similarity filtering within each scenario. This step provides a broad but controlled inventory of entities for subsequent annotation.

Stage B: Partonomy Construction. For each entity, we annotate a compact but comprehensive partonomy graph. Each entity is decomposed into essential, non-overlapping parts that together cover the object. For every part, we additionally annotate connectivity relations and concise functional/structural descriptions that explain how the part is situated in the whole object. This representation serves as the structural backbone for all later attribute and affordance annotation.

Stage C: Physical Attribute Annotation. For each part, we annotate multiple plausible **physical** attributes (rather than a single fixed description), including geometry, size/thickness, local features, material, rigidity, durability, elasticity, surface, weight, and concise summary notes. The objective is to represent realistic diversity in how the same part may manifest in practice, while keeping each variant internally consistent.

Stage D: Physical Variant Composition. Part-level physical variants are combined into entity-level physical configurations. When combinatorics are manageable, all combinations are retained; when the product is too large, combinations are subsampled with coverage-oriented selection so that each part's alternatives are represented. This yields multiple physically grounded versions of each entity.

Stage E: State Attribute Annotation. Conditioned on each physical configuration, we annotate part-level **state** attributes: accessibility (visibility and availability), condition (moisture and temperature), internal state (e.g., empty/filled where applicable), and summary notes. Multiple plausible states are generated per part to reflect ordinary and less typical but still realistic conditions.

Stage F: State Variant Composition. We then combine part-level state variants into entity-level state configurations, again with bounded combinatorial growth. The result is a set of entity variants where each part has both physical and state descriptors, enabling downstream affordance annotation under explicit conditions.

Stage G: Functional Affordance Annotation. For each part in each entity variant, we annotate a diverse set of functional affordances grounded in the already-annotated physical and state attributes. Each affordance includes: (i) use condition, (ii) environment condition, (iii) attribute evidence (physical/state and whether best conveyed visually or textually), (iv) affordance description, (v) level annotation (one intended/normal use and graded emergency alternatives), (vi) recipient condition, (vii) concrete recipient examples, and (viii) failure cases. This design enforces condition-aware, recipient-aware, and failure-aware affordance descriptions rather than unconstrained brainstorming.

Stage H: Entity Assembly and Consistency Check. Finally, part-level outputs are reassembled into complete entity records. Only entities with complete part coverage are retained, ensuring structural consistency between early decomposition and final affordance annotations. The final artifact is a complete, readable dataset of entities, parts, attributes, and affordances.

Prompting Strategy. We use task-specific prompts for each stage, with strict schema constraints and explicit instructions to promote diversity, plausibility, and consistency. These prompts are carefully designed to minimize ambiguity, ensure machine-parsable outputs, and preserve descriptive richness. All annotations are generated by GPT-5.2 under human supervision and quality control. Before scaling up annotation, we perform iterative human prompt refinement to improve annotation quality and ensure the prompts reliably produce the desired outputs. After the full annotation process, we conduct quality assessment by sampling 0.1% of the resulting data. This evaluation shows a pass rate of approximately 98% under automatic LLM-judge assessment and 95% under human review. Here, quality is defined in terms of commonsense validity, groundedness, and factual consistency across the entire annotation pipeline. For more details about the annotations, please refer to Table 5.

C.2 Quality Verification of KB Construction

To assess the quality of the generated affordance knowledge base, we recruit six PhD students with computer science backgrounds to evaluate randomly sampled KB cases according to shared evaluation instructions.

Each sampled case is evaluated using three binary criteria:

1. **Commonsense validity**: whether the entity, part, attribute, and affordance descriptions are plausible under ordinary physical-world knowledge;
2. **Attribute groundedness**: whether each affordance is supported by the annotated physical and state attributes rather than by unstated properties; and
3. **Factual consistency**: whether the part descriptions, attributes, conditions, recipients, and failure cases are mutually consistent.

A case is counted as passing only when it satisfies all three criteria.

In the original audit, human reviewers assign passing judgments to approximately 95% of the sampled KB cases. We additionally apply the same rubric through an automated LLM-based quality-control stage, which yields a pass rate of approximately 98%. Human and automated judgments agree in over 90% of the audited cases. These results provide

sampled evidence that the KB annotations are generally plausible, grounded, and internally consistent; they should not be interpreted as exhaustive human verification of every annotation.

C.3 Core Hyperparameters Details

Hyperparameter	Value
Number of scenarios	8
Sampled entities per scenario	50
Semantic similarity threshold (entity deduplication)	0.85
Physical variants generated per part (target range)	2–3
State variants generated per part (target range)	2–3

Table 6: Core hyperparameters of data-generation settings.

Hyperparameter	Value
Maximum parts per entity (partonomy cap)	8
Maximum physical combinations per entity	8
Maximum state combinations per physical entity variant	6
Maximum final variants per original entity	48
Affordances generated per part	6
Affordance level structure	Normal 0 + Emergency 1–5

Table 7: Core hyperparameters of combinatorial and annotation controls for coverage and tractability.

Hyperparameter	Value
Generation temperature	0.7
Parallel generation (part/attribute/affordance stages)	up to 1024 workers
Incremental save interval (other generation stages)	up to 50 generations

Table 8: Core hyperparameters of inference-time settings for scalable generation and fault-tolerant processing.

C.4 Core Prompt Details

System Prompt for Entity Decomposition Generation

```
You are an expert at analyzing object structure and component parts.

**Task:** Annotate the partonomy graph for "{entity_name}" (found in {scenario}).

**Requirements:**
1. List ALL the common parts (both externally visible and internally hidden ones)
2. Limit to 8 parts maximum, can be less if the object is simple and parts are obvious
3. Your parts should better have explicit boundaries which divide the object's different function units or
 affordance mechanisms
4. Include components that may have useful affordances themselves (e.g. screws, batteries, etc.).
5. Parts must be non-overlapping and cover the whole object (e.g. a blade contains cutting edge, so it's
one part).
6. Imagine a specific entity when you annotate the parts, don't include any optional parts or uncertain
wording, all parts should be necessary and essential to the entity's function.
7. Parts with same function differing only by position or direction should be annotated as ONE part
8. For each part: specify connected parts and describe function/connection

**Schema:**
```

```
```json
{{
  "entity_name": "...",
  "parts": ["part1", "part2", ...],
  "relations": {{
    "part1": {{
      "connected_to": ["part2", "part3"],
      "connection": "Description of part1, its function, and connection"
    }},
    ...
  }}
}}
```

Example (prescription reading glasses):
```json
{{
  "entity_name": "prescription reading glasses",
  "parts": ["front_frame", "lenses", "hinge_mechanisms", "temple_arms", "temple_tips"],
  "relations": {{
    "front_frame": {{
      "connected_to": ["lenses", "hinge_mechanisms"],
      "connection": "Main rigid front structure; provides lens openings/retention geometry; provides hinge
 mounting points."
    }},
    "lenses": {{
      "connected_to": ["front_frame"],
      "connection": "Two optical elements seated/retained by the front frame."
    }},
    ...
  }}
}}
```

First imagine a specific entity when you annotate the parts, and then provide reasoning about its
structure and key parts, and then output the parts and relations JSON.
```

System Prompt for Physical Attributes Annotation

```
You are an expert at analyzing physical properties of object parts.

**Task:** Annotate physical attributes for the "{part_name}" of "{entity_name}".

**Context:**
- All parts: {all_parts}
- Connected to: {connected_to}
- Connection: {connection}

**Annotate these attributes:**
1. **Geometry & Shape:** shape, size, thickness (thin/medium/thick/NA), local_features
2. **Material & Structural:** material, rigidity (very rigid/rigid/semi-rigid/flexible/soft), durability (
very fragile/fragile/normal/sturdy/very sturdy), elasticity (non-elastic/springy/stretchable/very
stretchable), surface
3. **Mass:** weight (very light/light/moderate/heavy/very heavy)
4. **Others:** Other important attributes for affordance (1-2 sentences)
5. **Summary:** Comprehensive summary of all attributes

**Requirements:**
- Generate 2-3 physical attribute combinations for this part
- Each combination must be plausible, internally consistent, and diverse
- Include common variations (e.g., plastic vs steel)
- Include one unusual but plausible variation if it creates distinctive affordances
- All fields required; use "NA" only if truly not important for creative affordance
- Be assertive; avoid "might", "maybe", "could be"
- Ensure consistency (e.g., plastic -> lighter, steel -> heavier)

**Schema:**
```json
[
  {{
    "shape": "...",
    "size": "...",
    "thickness": "...",
    "local_features": "...",
    "material": "...",
    "rigidity": "...",
```
```

```
    "durability": "...",
    "elasticity": "...",
    "surface": "...",
    "weight": "...",
    "others": "...",
    "summary": "..."
  }}
]
```

Example (glasses front_frame):
```json
[
  {{
    "shape": "two connected loops and narrow bridge",
    "size": "hand-held",
    "thickness": "thin",
    "local_features": "two enclosed openings and flat front edge and narrow bridge span",
    "material": "plastic",
    "rigidity": "rigid",
    "durability": "normal",
    "elasticity": "non-elastic",
    "surface": "smooth",
    "weight": "very light",
    "others": "continuous rim can hook or hang on thin supports; bridge provides pinchable grip point",
    "summary": "A lightweight thin rigid plastic double-loop rim with enclosed openings and a pinchable
bridge, smooth surfaced and easy to hang or hold."
  }},
  {{
    "shape": "two connected loops and narrow bridge",
    "size": "hand-held",
    "thickness": "thin",
    "local_features": "metal rim edge and small screw openings",
    "material": "steel",
    "rigidity": "very rigid",
    "durability": "sturdy",
    "elasticity": "non-elastic",
    "surface": "smooth and slightly cool",
    "weight": "light",
    "others": "conductive metal edge can transfer heat or cold; thin rim can fit into narrow slots",
    "summary": "A thin sturdy metal double-loop rim with small openings and a narrow bridge, rigid and
thermally conductive with smooth edges."
  }}
]
```

First provide reasoning. Then output JSON array of 1-5 combinations.

System Prompt for State Attributes Annotation

```
You are an expert at analyzing state and condition of object parts.

**Task:** Annotate state attributes for the "{part_name}" of "{entity_name}" with given physical
attributes.

**Physical Attributes:**
{physical_attrs}

**Annotate these state attributes:**
1. **Access State:**
   - Visibility: visible, partially visible, hidden (relative to whole entity)
   - Availability: free, partially blocked (easily freed by hand), blocked (requires tools)
2. **Condition State:**
   - Moisture: dry, slightly wet, wet, NA
   - Temperature: cold, slightly cold, slightly hot, hot, NA (room temp)
3. **Internal State:**
   - Internal: empty, partially filled, full, NA (physical or abstract capacity)
4. **Others:** Other important state attributes (1-2 sentences)
5. **Summary:** Comprehensive summary of all state attributes

**Requirements:**
- Generate 2-3 state attribute combinations for this part
- Must be consistent with physical attributes
- Include common/typical state and unusual but plausible states
- All combinations must be plausible, consistent, and diverse
- All fields required; use "NA" if not important
```

```
- Be assertive; avoid "might", "maybe"
- States should not contradict physical attributes

**Schema:**
```json
[
  {{
    "visibility": "...",
    "availability": "...",
    "moisture": "...",
    "temperature": "...",
    "internal": "...",
    "others": "...",
    "summary": "..."
  }}
]
```

Example (vacuum bag - flexible, dry):
```json
[
  {{
    "visibility": "hidden",
    "availability": "partially blocked",
    "moisture": "dry",
    "temperature": "NA",
    "internal": "partially filled",
    "others": "Seated inside a closed compartment; compartment cover is latched shut; bag collar aligned
on plastic inlet mount; bag material flexible but holds shape from airflow.",
    "summary": "Hidden in latched compartment and partially blocked by cover and inlet mount; dry at room
temperature; partially filled; flexible bag seated on collar mount."
  }},
  {{
    "visibility": "hidden",
    "availability": "blocked",
    "moisture": "slightly wet",
    "temperature": "NA",
    "internal": "partially filled",
    "others": "Contents clumped and tacky, bag adheres to compartment liner; bag surface damp and slightly
 softened; collar stuck on inlet mount and won't release with simple pull.",
    "summary": "Hidden and blocked by adhesion and stuck collar mount; slightly wet at room temperature;
partially filled with clumped debris preventing easy hand removal."
  }}
]
```

First provide reasoning considering physical attributes. Then output JSON array of 1-4 combinations.
```

System Prompt for Affordance Generation

```
You are an expert at identifying functional affordances based on attributes.

**Task:** Annotate functional affordances for the "{part_name}". This part belongs to the entity "{
entity_name}" in the scenario "{scenario}".

**Physical Attributes of Part "{part_name}":**
{physical_attrs}

**State Attributes of Part "{part_name}":**
{state_attrs}

**Instructions:**
Identify 6 different functional affordances for this part. For each:

1. **Use Condition:** What preparation is needed to access this affordance?
   - "NA" if part is free and directly usable
   - Otherwise describe preparation steps (e.g., "break the lens", "remove from case")
   - Based on visibility/availability from state attributes

2. **Environment Condition:** What environmental conditions are needed for this affordance?
   - Focus on scenario/environment requirements (e.g., "lighting available", "power source nearby")
   - NOT about the part itself or recipient, but about external conditions
   - "NA" if no special environment needed

3. **Attribute:** What attributes enable this affordance?
```

```
   - List relevant attributes that are needed to be considered for this affordance; your stated attributes
 must be derived from the given physical + state attributes above; do not introduce new attributes that
are not provided
   - Format: JSON list of lists `[["attribute statement", "physical/state", "visual/text", "explanation
why visual/text"], ...]`
   - "physical" = attribute statement derived from physical attributes given above; "state" = attribute
statement derived from state attributes given above
   - "visual" = this attribute can be clearly illustrated in an image without ambiguity; "text" = this
attribute needs text description to clarify (e.g., temperature, texture, hidden features)
   - The explanation should be your reason why this attribute needs to be illustrated using text to
clarify or only visual signal is enough to convey the information.
   - Example: `[["the material of the lens is glass", "physical", "visual", "Glass material is visually
identifiable by its transparency and reflective surface"], ["the surface is smooth", "physical", "text", "
Smoothness is a tactile property difficult to convey visually alone"]]`

4. **Affordance:** What can this part be used for?
   - Brief, clear description of the function/purpose
   - Can be original normal function (must have at least one original normal function) OR creative
alternative use
   - Must be plausible and grounded in this part's given physical and state attributes above
   - Must act upon a recipient to take effect: passive or decorative roles (e.g., "use as decoration", "
display for aesthetics") are NOT valid affordances since they do not act upon anything
   - Write at a high, general level: describe the functional capability, not a specific scenario (good: "
dig into soft material"; bad: "use as a shovel in a garden").
   - If the use is too rare, niche, or uncommon to be practically meaningful, do not include it.
   - If the affordance requires specific conditions to work, those must be reflected faithfully in
use_condition, environment_condition, and recipient_condition

5. **Level:** Categorize the affordance type and suitability
   - "Normal 0" = Original intended normal function of the part within the entity (must have at least one
original normal function)
   - "Emergency 1-5" = Creative/alternative use
     - 1 = Rarely used, only if nothing else available; difficult to access or imperfect; in real life
people rarely use this affordance this way
     - 2 = In between 1 and 3
     - 3 = Moderately suitable, could work in urgent situations; in real life people may use this
affordance a few times in real emergencies but not too much
     - 4 = In between 3 and 5
     - 5 = Highly suitable, easy and natural to use; very effective; in real life people are willing to
use this affordance this way most of the time
   - Take into consideration if normally people can have access to this affordance, whether this part
needs to be detached from the whole entity to use this affordance, whether to use this affordance will
irreversibly damage the part or the whole entity, and how likely it is to happen, etc.
   - Format: "Normal 0 / Emergency X (comprehensive and grounded reasoning why you annotate this certain
level)"

6. **Recipient Condition:** What attributes must the recipient have?
   - Every affordance must have a recipient. It can be the object, person, or material that this part acts
 upon
   - Define scope and limits using attribute categories (shape, size, rigidity, durability, surface, etc.)
   - Be fine-grained and comprehensive
   - Example: "thin to medium thickness, soft to semi-rigid rigidity, not harder than glass"

7. **Example Recipient:** List 3-4 concrete examples
   - Must satisfy the recipient condition
   - The proposed recipient must be concrete and specific things or objects that can be easily found in
real life, rather than abstract concepts or ideas
   - Choose diverse examples reflecting affordance scope

8. **Failure Case:** When will this affordance NOT work?
   - Comprehensively consider all failure situations:
     * Use condition failures (can't access/prepare the part)
     * Environment condition failures (lack of necessary environmental factors)
     * Recipient condition failures (recipient too hard/thick/incompatible)
     * Action condition failures (user lacks skill/force/precision)
     * Other practical limitations
   - Be specific and realistic

**Requirements:**
- Generate 6 diverse and non-overlapping affordances
- Generate exactly one affordance for each level (Normal 0 and Emergency 1-5)
- Keep everything plausible and grounded-realistic physical-world uses only
- Each affordance should be completable solely by this part of the entity, NOT relying on other parts or
the whole entity
- Do not introduce attributes not provided above, and just focus on this part's attributes, NOT other
parts' attributes or the whole entity's attributes
- Every affordance must act upon a recipient; please skip any purely decorative, passive, or static role
- Describe the affordance at a high, scenario-agnostic level; omit uses that are too rare or uncommon to
be practically meaningful
```

```
- If an affordance requires specific conditions (use, environment, recipient), those conditions must be
explicitly captured in the corresponding fields

**Schema:**
```json
[
  {{
    "use_condition": "...",
    "environment_condition": "...",
    "attribute": [["attribute statement", "physical/state", "visual/text", "explanation why visual/text"],
 ...],
    "affordance": "...",
    "level": "Normal 0 / Emergency 1-5 (reason)",
    "recipient_condition": "...",
    "example_recipient": "...",
    "failure_case": "..."
  }}
]
```

Example (glass lens, part of glasses entity - visible, free, dry, transparent):
```json
[
  {{
    "use_condition": "NA",
    "environment_condition": "NA",
    "attribute": [
      ["the material of the lens is glass", "physical", "visual", "Glass is visually identifiable by
transparency and clarity"],
      ["the shape is round and flat disc", "physical", "visual", "Circular flat shape is directly
observable"],
      ["the rigidity is very rigid", "physical", "text", "Rigidity requires text description to confirm as
 there is no visual signal to indicate rigidity"]
    ],
    "affordance": "redirect, focus, or magnify light for vision correction or reading",
    "level": "Normal 0 (original intended function of the lens in the glasses)",
    "recipient_condition": "light must pass through; viewer needs magnification; text or objects at
appropriate focal distance",
    "example_recipient": "printed text in books, small labels, fine details on objects, digital screens",
    "failure_case": "Fails if lens is dirty or scratched (blocks light), wrong prescription (incorrect
magnification), insufficient lighting, or recipient is beyond focal range"
  }},
  {{
    "use_condition": "If we break the glass lens into pieces first",
    "environment_condition": "NA",
    "attribute": [
      ["the material of the lens is glass", "physical", "visual", "Glass is visually identifiable by
transparency and clarity"],
      ["the durability is fragile", "physical", "text", "Fragility isn't directly visible, requiring text
description to confirm"],
      ["the edge becomes sharp after breaking", "physical", "visual", "The sharpness after of broken
glasses is commonsense knowledge, so no need to mention it through text description"]
    ],
    "affordance": "cut, scrape, or pierce small items",
    "level": "Emergency 2 (rarely used due to danger; people rarely break the glasses lens only for
cutting purposes; it needs get the lens out first and break it into pieces is irreversible, make itself
and the whole glassess not usable anymore; only when no proper cutting tool available)",
    "recipient_condition": "thin to medium thickness, soft to semi-rigid rigidity, not harder than glass,
not highly abrasive",
    "example_recipient": "tape, paper, thin plastic wrap, soft fruit skin",
    "failure_case": "Fails if recipient too hard (damages edge), too thick (can't penetrate), or user can'
t safely handle sharp glass (risk of cuts); fails if pieces too small to grip"
  }},
  ...
]
```

First provide reasoning about affordances. Then output JSON array of 6 diverse and non-overlapping
affordances of different levels following the schema and instructions above.
```

D Task Creation Pipeline Details

D.1 Overview of Sampling Stages

The task creation pipeline is designed to produce rigorous yet creativity-demanding benchmark tasks. For each task, we enforce three properties: (1) a clearly defined gold affordance (entity-part-action) that is truly preferred, (2) distractor entities that are controlled by semantic similarity and decision ambiguity, and (3) a grounded first-person task narrative with judge-checkable solution constraints. We organize the pipeline into conceptual stages as follows.

Stage A: Scenario-wise Semantic Clustering. We first aggregate all annotated affordances (with their part-level physical/state context) and build two lookup spaces: an entity-level lookup and an affordance-level lookup. This allows every sampled affordance to be traced back to its exact scenario, entity, part, and annotation fields. Then, within each scenario, we embed affordance texts using Text-Embedding-3-Large and perform complete-linkage hierarchical clustering with an adaptive distance threshold plus a hard diameter cap. This yields compact semantic bins of affordances and supports structured gold sampling by (i) cluster-size band and (ii) affordance level (Normal 0 or Emergency 1–5). In our current run, this produces roughly 3.2K–3.7K clusters per scenario.

Stage B: Stratified Gold Affordance Sampling. A gold affordance is sampled under explicit controls: cluster-size range (rarity/commonness proxy) and affordance level (normal vs. emergency tiers). This stratification makes the benchmark composition interpretable and analyzable across controlled factors, rather than being purely random.

Stage C: Task Prompt Instantiation from Gold. Given the sampled gold affordance, we generate a first-person task description with a concrete recipient and recipient condition, while hiding the gold entity/part/mechanism. This step ensures the task is goal-centric ("what should I use and how?") instead of answer-leaking.

Stage D: Intra-Entity Gold Dominance Self-Check. Before accepting the sampled gold, we compare it against **other parts of the same entity**. The gold is rejected and resampled if another part can challenge it strongly (e.g., replacement judged possible, or high-ambiguity similarity). This gate ensures the chosen gold is internally dominant within its own entity.

Stage E: Candidate Noise Pool Sampling. After a gold passes self-check, we sample candidate non-gold entities (noise entity candidates) using embedding-distance heuristics: a near set (semantically close distractors) and a far set (semantically dissimilar distractors). This supports controlled distractor composition from "highly confusable" to "clearly different."

Stage F: Inter-Part Comparison Against Gold. For each sampled entity, every part is judged against the gold using its physical/state attributes and existing affordances. The judge outputs: (1) whether a similar affordance is feasible, (2) a structured affordance annotation (if feasible), (3) whether this should replace the gold, and (4) decision-making difficulty (1–5). Replacement decisions explicitly consider accessibility, consequences, willingness, commonness, and safety.

Stage G: Distractor Filtering and Difficulty Assignment. Entities are filtered by strict rigor rules: if any part is judged better than gold, or ambiguity is high, the entity is excluded. Remaining entities are split into: `not_similar` (no similar affordance) and `similar` (similar but not better). We then form different tiers: Dissimilar (all not_similar), Mixed (balanced mix), and Similar (all similar-but-not-better), with controlled entity counts.

Stage H: Final Task Assembly. For each accepted case, we assemble: gold annotation, selected entities, judge outputs, additional scene items, first-person environment description,

and a structured four-step solution. Solution fields are aligned with recipient/use/environment conditions plus affordance application mechanics, enabling direct judge-side verification.

Prompting Strategy. Similarly, we use stage-specific prompts with strict JSON schema constraints and explicit instructions to enforce groundedness, comparability, and decision consistency across sampling, comparison, and final task composition. The prompts are designed to reduce ambiguity, avoid gold leakage, and keep all outputs machine-parsable while preserving realistic first-person task narratives. All data are generated with GPT-5.2 under human supervision and iterative prompt refinement before full-scale runs. During refinement, we repeatedly inspect pilot outputs and update prompts to correct common failure modes (e.g., weak gold-dominance checks, inconsistent replacement decisions, or under-specified conditions). After large-scale generation, we perform quality assessment by sampling a subset of 0.1% outputs and verify the logical coherence, condition-grounded reasoning, and consistency between gold selection. The generated data passes 98% human checks and reaches the bar of high quality.

D.2 Quality Verification of Task Construction

To assess the quality of the generated benchmark tasks, we recruit six PhD students with computer science backgrounds to evaluate randomly sampled tasks according to shared evaluation instructions.

Each sampled task is evaluated using three binary criteria:

1. **Solution logical coherence**: whether the annotated solution can plausibly accomplish the stated task;
2. **Attribute-grounded rationale**: whether the proposed mechanism is supported by the gold part's physical and state attributes and satisfies the stated use, environment, and recipient conditions; and
3. **Gold-comparison consistency**: whether the annotated reasons for rejecting the alternative entity–part combinations are consistent with the final selection of the gold solution.

A task is counted as passing only when it satisfies all three criteria.

In the original sampled audit, approximately 98% of tasks pass all three criteria. We subsequently expand the human audit to 285 tasks, corresponding to approximately 2% of the 14,280-task benchmark and covering different task settings and entity types. The overall pass rate remains above 96%, while the gold-comparison consistency criterion achieves a pass rate of 96%. These results indicate that the observed construction quality remains stable under the expanded audit, although residual ambiguity may remain in a small proportion of cases.

D.3 Core Hyperparameters Details

Hyperparameter	Value
Embedding model	Text-Embedding-3-Large
Clustering method	Complete-linkage hierarchical clustering
Hard max cluster diameter (cosine distance)	0.35
Gold level set	Normal 0 and Emergency 1–5
Gold cluster-size bands	(2,4), (5,10), (10,50)
Cases per (level, cluster-size) tier	10

Table 9: Core hyperparameters of gold-sampling and clustering controls.

Hyperparameter	Value
Candidate entities sampled per comparison set	30
Distractor-count settings in final tasks	3, 6, 9, 12
Affordance similarity tiers	Dissimilar, Mixed, Similar
Mixed-tier composition	50% Similar + 50% Dissimilar
Cases per (similarity, entity-count) tier	2

Table 10: Core hyperparameters distractor sampling and final-task composition controls.

Hyperparameter	Value
Comparison judge temperature	0.0
Task generation and final composition temperature	0.3
Current final tasks generated	14,280

Table 11: Core hyperparameters inference settings and resulting scale of the current run.

D.4 Core Prompt Details

System Prompt for Gold-to-Task Generation

```
You are writing a first-person benchmark question for a creative problem-solving task. Given a gold
affordance, your job is to invent a concrete, realistic situation that the affordance can solve - without
revealing the gold entity, part, or affordance mechanism.

Gold Entity: {entity_uid}
Gold Part: {part_name}
Gold Affordance:
{affordance}

**Your process:**
Step 1 - Pick a concrete recipient from `example_recipient` (or one that satisfies `recipient_condition`).

Step 2 - Invent a specific real-world situation that person encounters involving that recipient.
Step 3 - Describe the situation and goal in everyday language, then ask "What can I use?" or "What should
I use and how?".

**Rules:**
- Start from the SITUATION and RECIPIENT, not from the affordance mechanism. Ask yourself: "What real
problem is this person facing?" not "What does the affordance do?"
- Be specific and concrete. Name the recipient object. Give context (where, what happened, what they want
to achieve).
- Do NOT paraphrase or echo the affordance description. If the affordance is "dampen a surface", do NOT
write "make a spot wet". Describe WHY the person needs it.
- Do NOT name or hint at the gold entity or part.
- Do NOT describe what needs to happen physically - describe the human goal/problem.

**Examples (affordance: "apply small amount of moisture to a surface"):**
```

```
GOOD: "My fabric jacket has a small stubborn stain from yesterday's lunch. I want to treat just that spot
before tossing it in the wash, but I don't have a spray bottle or wet cloth nearby. What can I use?"
  -> Concrete: specific object (fabric jacket), specific problem (stain), specific constraint (no wet
cloth). Affordance is not mentioned.

BAD: "I'm trying to quickly put a small, localized damp spot onto an absorbent surface. What can I use?"
  -> Describes the affordance mechanism directly. No real situation. Not grounded.

BAD: "I need to use the sponge tip to apply moisture to my shirt." -> Leaks entity and mechanism.

Output JSON:
```json
{{
  "task": "Concrete first-person scenario question grounded in a specific recipient and situation, hiding
the gold entity/part/affordance",
  "recipient": "The specific object or thing receiving this affordance (pick from example_recipient or
invent one matching recipient_condition)",
  "recipient_condition": "Required state or attribute of the recipient"
}}
```
```

System Prompt for Gold-vs-Target Part Judge

```
You are judging whether a target part can fulfill a similar role to a gold affordance for a given task.

## Gold
Entity: {gold_entity} | Part: {gold_part}
Physical Attributes of Part {gold_part}:
{gold_physical_attrs}
State Attributes of Part {gold_part}:
{gold_state_attrs}
Gold Affordance:
{gold_affordance}

## Target Part
Entity: {target_entity} | Part: {target_part}
Physical Attributes of Part {target_part}:
{physical_attrs}
State Attributes of Part {target_part}:
{state_attrs}
Existing Affordances of Part {target_part}:
{existing_affordances}

## Task
{task}
Recipient: {recipient}

---

**Step 1 - Similarity Judgment**
Determine whether this part ALONE (without any other part of the entity) can perform a functionally
similar role to solve the task - i.e., it serves the same purpose as the gold affordance. Base your
judgment strictly on the part's provided physical and state attributes.

**Step 2 - Affordance Annotation** (only if similar_affordance = Yes)
If an existing affordance already matches, adapt it. Otherwise write a new one. All fields must be
grounded in the provided attributes. If similar_affordance = No, set ALL affordance fields to "NA".

Field definitions:
- use_condition: preparation needed to access this affordance; "NA" if part is free and directly usable (
based on visibility/availability from state attributes)
- environment_condition: external environmental conditions required (not about the part or recipient); "NA
" if none
- attribute: list of [["attribute statement", "physical/state"]], indicating all given target part
attributes enabling this affordance;
- recipient_condition: required attributes of the recipient (shape, size, rigidity, material, etc.), fine-
grained
- example_recipient: 3-4 concrete examples satisfying recipient_condition
- failure_case: all situations where this affordance fails - use condition failures, environment failures,
 recipient incompatibility, action/skill failures

**Step 3 - Gold Comparison** (only if similar_affordance = Yes)
Systematically compare gold vs. target across ALL of the following aspects. For each aspect, explicitly
state which side is better and by how much:
- Accessibility and use/env conditions (how easy to access and activate)
- Future consequences (irreversible damage, side effects, mess)
```

```
- How willing people are to use it this way (social acceptability, effort)
- How common this usage is in everyday life
- Safety and ethical considerations

After comparing all aspects, make a final decision. If most aspects clearly favor one side, gold_change is
 straightforward. If several aspects are roughly equal, or pros and cons are genuinely hard to weigh
against each other, explicitly acknowledge that uncertainty and reflect it with a high
decision_making_difficulty score. If similar_affordance = No, copy the Step 1 reason into
gold_change_reason and set gold_change = No.

Output JSON:
```json
{{
  "similar_affordance_reason": "whether this part ALONE can show similar affordance to solve the task,
based on which attributes and conditions",
  "similar_affordance": "Yes or No",
  "affordance": {{
    "use_condition": "...",
    "environment_condition": "...",
    "attribute": [["attribute statement", "physical/state"]],
    "affordance": "...",
    "recipient_condition": "...",
    "example_recipient": ["...", "...", "..."],
    "failure_case": "..."
  }},
  "gold_change_reason": "For each aspect - accessibility, consequences, willingness, commonness, safety -
state which side is better and why. Then give a final balanced verdict. If aspects conflict or are too
close to call, say so explicitly.",
  "gold_change": "Yes or No",
  "decision_making_difficulty": "1-5 or NA"
}}
```
decision_making_difficulty: 1 = very easy to keep gold (one side clearly better across most aspects); 5 =
extremely difficult (aspects are close or contradictory, genuinely uncertain which is better); NA if
similar_affordance is No. A high score (4-5) is appropriate whenever multiple aspects are roughly tied or
point in opposite directions - do not default to low scores when the comparison is genuinely hard.
```

System Prompt for Final Task Packaging

```
Generate three fields for a benchmark task.

## Gold
Entity: {gold_entity} | Part: {gold_part}
Gold Affordance:
{gold_affordance}

## Task
{task}
Recipient: {recipient} (Condition: {recipient_condition})

## Entities already in the scene
{entities_desc}

---

**items**: List things referenced in the gold affordance's use_condition, environment_condition, and
recipient_condition, plus the recipient itself. Describe each with the traits implied by those conditions.
 Also add 3-5 natural noise objects that fit the {scenario} but are unrelated to the task (interactable=No
). Do NOT add things similar to the entities above.

**environment**: First-person scene description starting "I am in the {scenario}. Around me there is ..."
- naturally mention every entity and key item.

**solution**: A structured dict with exactly four keys describing how to solve the task using {gold_entity
}'s {gold_part}. Each value is a string in the format "... (Note: ...)" where the Note contains specific
things a judge should explicitly verify when scoring. If no specific condition was explicitly needed,
write "(Note: NA)".

The four keys and what to write for each:
- "prepare_recipient": How to prepare the recipient to be applied for the task.
- "prepare_use_condition": How to set up the tool (gold part) to meet use_condition.
- "prepare_environment_condition": What environmental setup is needed to meet environment_condition.
- "apply_affordance": The core action about how to apply {gold_part}'s affordance to the recipient to
solve the task. This is the most important step. Be comprehensive and detailed, referencing the specific
physical/state attributes of the part that make this work.
```

```
Rules:
- Please be strictly grounded in the gold affordance annotation and the part's listed attributes. Do not
invent steps or attributes beyond what is annotated.
- Each step should be detailed and concrete, but not overly complex.
- The Note in each step is a judging reference.

Output ONLY valid JSON:
```json
{{
  "items": [
    {{
      "name": "<item name>",
      "description": "<description; include condition-relevant attributes where applicable>",
      "interactable": "Yes or No"
    }}
  ],
  "environment": "<I am in the {scenario}. Around me there is ...>",
  "solution": {{
    "prepare_recipient": "<description of recipient preparation> (Note: <critical judging reference of
recipient_condition attributes that need additional verification besides that in the task description, or
NA>)",
    "prepare_use_condition": "<description of tool setup> (Note: <critical judging reference of
use_condition that needed to be set up before the use, or NA>)",
    "prepare_environment_condition": "<description of environment setup> (Note: <critical judging
reference of environment_condition that needed to be set up before the use, or NA>)",
    "apply_affordance": "<detailed core action using {gold_part} on the recipient> (Note: <critical
judging reference of the key attributes and affordance mechanism to verify before applying the affordance
>)"
  }}
}}
```
```

E Experiment Details

E.1 Evaluation Settings, Metrics, and Judge Prompt

Settings. The main results are reported under a standardized decoding setup designed to reduce variance across models. Most models are evaluated with **temperature 0**, so outputs are as deterministic as possible and differences more directly reflect model capability rather than sampling noise. Two exceptions are **GPT-5-Mini** and **GPT-5-Nano**, which do not expose an adjustable temperature parameter; for these models, we use the default decoding setting provided by the API. Aside from this constraint, all models are evaluated under the same prompting and output-format requirements.

To avoid truncation affecting performance, all models are assigned a maximum generation budget of at least **16K output tokens**, which we found sufficient for the full task, including tool selection, part-level justification, and action description. In the main setting, each model receives the task description, the full list of entities in the scene, and the annotated attributes of their parts, and is asked to identify the most suitable entity and part and explain how they should be used to solve the task. Unless otherwise noted, each task is evaluated with a single generation per model. The exact prompt template used for the main setting is provided below.

System Prompt for Final Task Packaging

```
TASK:
{task content}

ENVIRONMENT:
{environment content}

=== ENTITIES AVAILABLE (full descriptions) ===
{blocks of entity description, from task annotation}

=== OTHER ITEMS IN SCENE ===
{blocks of other items description, from task annotation}
```

```
Identify the best entity part to creatively accomplish the task. Please first provide your reasoning
process, and then finally respond with a JSON object containing the three answer fields:
<your reasoning process here>
{"gold_entity": "<entity name, case-sensitive and exact match>", "gold_part": "<part name, case-sensitive
and exact match>", "how_to_use": "<detailed instructions>"}
```

Metrics. In the main setting, we report two objective metrics, *gold correct* and *entity correct*, as well as six subjective metrics. For each metric, our original scoring is between 0–2. However, considering the alignment with the scale we employ in Section 3, we uniformly scale the range to 1–5. In the results reported in the main table, the subjective metrics are only used for judging those answers that count as Gold Correct, while leaving other failure cases' analysis in the later section. Consistent with Section 3, the subjective metrics evaluate aspects such as constraint coverage, physical grounding, and related properties of the predicted affordance. In particular, these subjective metrics are designed to assess the quality of the model's generated "how to use" explanation, rather than only whether the correct entity and part are selected.

For subjective evaluation, we use Gemini-3.1-Flash-Lite as the judge model. The judging prompt is provided below.

System Prompt for Final Task Packaging

```
You are a strict evaluator for embodied task instructions.
Evaluate whether the predicted "how_to_use" is feasible compared to the gold affordance and gold solution.
Use evidence-based judgments for each field. Please follow the rubric strictly.

Task:
{task content}

Gold affordance JSON:
{gold affordance}

Gold solution JSON:
{gold solution}

Predicted how_to_use:
{how to use part of prediction}

Field-by-field rubric when judging the predicted "how_to_use":
1) environment_condition_covered (0/1/2/"NA"):
- 0: required external environment setup in gold is not mentioned at all in the predicted "how_to_use".
- 1: required external environment setup in gold is mentioned, but not all are covered in the predicted "
how_to_use".
- 2: required external environment setup in gold is covered well and reasonable in the predicted "
how_to_use".
- NA: only if in the gold affordance annotation, the environment_condition is NA.

2) use_condition_covered (0/1/2/"NA"):
- 0: required preparation/access of the tool-part is not mentioned at all in the predicted "how_to_use".
- 1: required preparation/access of the tool-part is mentioned, but not all are covered in the predicted "
how_to_use".
- 2: required preparation/access of the tool-part is covered well and reasonable in the predicted "
how_to_use".
- NA: only if in the gold affordance annotation, the use_condition is NA.

3) recipient_condition_covered (0/1/2/"NA"):
- 0: recipient-side prerequisites are not mentioned at all in the predicted "how_to_use".
- 1: recipient-side prerequisites are mentioned, but not all are covered in the predicted "how_to_use".
- 2: recipient-side prerequisites are covered well and reasonable.
- False: recipient assumptions are not met in the predicted "how_to_use".
- NA: only if in the gold affordance annotation, the recipient_condition is NA.

4) attributes_grounding (0/1/2): [compare to the key enabling attributes]
- 0: the predicted action is not grounded in the key enabling attributes of the gold affordance, or it
violates some attributes of the part.
- 1: the predicted action is mostly grounded in the key enabling attributes of the gold affordance, but
not all are covered or implied.
- 2: the predicted action is fully grounded in the key enabling attributes of the gold affordance, and
most of them are explicitly mentioned, covered or implied.

5) prediction_correctness (0/1/2): [compare to the gold solution]
- 0: compared with the gold solution the predicted "how_to_use" is completely wrong, not working at all.
```

```
- 1: compared with the gold solution the predicted "how_to_use" is partially workable but missing crucial
details/order/precision.
- 2: compared with the gold solution the predicted "how_to_use" is operationally correct and complete,
almost completely aligned with gold solution.

6) action_feasibility (0/1/2): [not compare to the gold solution, but focus on the action itself]
- 0: the action itself is physically impossible, not working at all, very unlikely to be used in practice,
 or unsafe.
- 1: the action itself is partially workable but still there are some steps that are not plausible,
aligned with common sense or not feasible.
- 2: the action itself is operationally correct and complete, almost completely aligned with common sense
and feasible in practice.

Evidence policy:
- Every *_reason must quote concrete evidence from predicted "how_to_use" and other given relevant
information.
- If evidence is unclear/missing, default to a stricter outcome (lower score).
- Make your reasoning clear, concise, and to the point for each field, and your score should be based on
the evidence.

Please make sure to only return a valid JSON with exactly these 12 fields (no extras, no markdown).
{{
    "environment_condition_covered_reason": "...",
    "environment_condition_covered": 0/1/2/"NA",
    "use_condition_covered_reason": "...",
    "use_condition_covered": 0/1/2/"NA",
    "recipient_condition_covered_reason": "...",
    "recipient_condition_covered": 0/1/2/"NA",
    "attributes_grounding_reason": "...",
    "attributes_grounding": 0/1/2,
    "prediction_correctness_reason": "...",
    "prediction_correctness": 0/1/2,
    "action_feasibility_reason": "...",
    "action_feasibility": 0/1/2,
}}
```

E.2 Reliability of the LLM-Based Evaluation

We use both objective metrics and LLM-judged metrics in our evaluation. We clarify the role of these metrics and provide an additional post-hoc validation of the LLM judge.

Objective primary metrics. **Gold Correctness** and **Entity Correctness** are exact-match metrics that do not rely on LLM-based evaluation. Gold Correctness measures whether both the predicted entity and part match the annotated gold answer, while Entity Correctness measures whether the predicted entity matches the gold entity. These two metrics serve as the primary metrics in our main experiments and subsequent analyses.

The remaining metrics, including constraint coverage, physical grounding, action feasibility, and overall correctness, are evaluated using an LLM judge. These metrics provide additional analysis of the quality of the predicted tool usage.

Constrained rubric-based judging. We use Gemini-3.1-Flash-Lite as the LLM judge. The evaluation follows a rubric- and rule-based protocol in which each score level is explicitly defined. The annotated gold solution is provided as an anchor for comparison, rather than allowing the judge to perform an unconstrained assessment. The complete evaluation prompt and scoring rubric are provided in Section E.

Under this design, the LLM judge primarily performs structured instruction following according to predefined evaluation criteria. All LLM judging is conducted with temperature set to 0 to ensure deterministic evaluation.

Potential model-family bias. Using a Gemini-family model as the judge does not appear to systematically favor Gemini-family models. As shown in **??**, Gemini models do not obtain the highest scores on the objective Gold Correctness and Entity Correctness metrics. These metrics are calculated through exact matching and are independent of the LLM judge.

We also use LLMs during KB and task construction. However, task construction and task evaluation serve different purposes. The KB and task construction pipelines are decomposed

Metric	Agreement	95% CI Lower	95% CI Upper
Human–original agreement	90.40%	89.90%	90.90%
Inter-human agreement	87.00%	86.70%	87.30%
GPT–original agreement	93.40%	91.90%	94.70%

Table 12: Agreement between the original Gemini-3.1-Flash-Lite judge, human judges, and an additional GPT-5.2 judge on 200 responses balanced across models and evaluation settings.

into fine-grained stages, each governed by detailed prompts and explicit instructions, as described in Sections C and D. Similarly, the evaluation judge follows an explicit rubric rather than performing unconstrained judgment.

Empirically, GPT- and Gemini-family models do not outperform Qwen on the objective metrics in **??**, even though GPT- and Gemini-family models are used during benchmark construction or evaluation. This result suggests that the benchmark does not simply favor the model families involved in its construction and evaluation.

Post-hoc validation setup. To further evaluate the reliability of the LLM-based evaluation, we conduct an additional post-hoc study using human judges and an additional LLM judge. We randomly select 200 model responses, balanced across models and evaluation settings.

For human evaluation, we recruit 10 students with computer science backgrounds. The human judges evaluate the selected responses using the same prompt instructions as the original judge. We also use GPT-5.2 as an additional LLM judge under the same prompt instructions. All LLM judging is conducted with temperature set to 0.

We report three agreement measures:

1. agreement between the human judgments and the original judge;
2. agreement among the human judges; and
3. agreement between GPT-5.2 and the original judge.

Results. As shown in Table 12, agreement between the human judgments and the original judge is 90.40%, with a 95% confidence interval of $[89.90\%, 90.90\%]$. Inter-human agreement is 87.00%, with a 95% confidence interval of $[86.70\%, 87.30\%]$. Agreement between GPT-5.2 and the original judge is 93.40%, with a 95% confidence interval of $[91.90\%, 94.70\%]$.

These results show strong consistency among the original judge, human judges, and the additional GPT-based judge under the same evaluation instructions. Together with the use of objective exact-match primary metrics and the constrained rubric-based judging protocol, these results support the reliability of the LLM-based auxiliary evaluation.

F Qualitative Examples

We here provide a step by step example showing how a task is formed and constructed. The example shows the full grounding chain: scenario → entity → partonomy → physical/state attributes → affordance → recipient → task → conditions → solution.

System Prompt for Final Task Packaging

```
0. Input Scenario and Entity

The annotation pipeline starts from a seed entity and scenario.

Scenario: `home_office`
Seed entity: `tape dispenser with weighted base`
```

```
This means all later parts, attributes, and affordances are grounded in an object that plausibly appears
in a home office.

1. Entity Partonomy Construction

The entity is decomposed into functional parts. For this entity, example parts include:

`housing_frame`
`weighted_base_insert`
`tape_roll_spindle`
`tape_brake_pad`
`cutter_blade`
`non_slip_bottom_pad`
...

The gold part selected later is `housing_frame`.

2. Part Attribute Annotation

Each part receives physical and state attributes. For the gold part `housing_frame`:

Physical attributes: rigid ABS shell; internal ribs and bosses; molded bearing pockets; boxy internal
skeleton.
State attributes: visible, exposed, reachable, and freely handled.

These attributes are important because the task is not generated from the object name alone. The later
affordance is grounded in concrete part-level properties.

3. Functional Affordance Annotation

The gold part receives affordances based on those attributes.

Gold entity: `tape dispenser with weighted base`
Gold part: `housing_frame`
Gold affordance: use the molded pockets/posts to locate and align cylindrical or rod-like objects, acting
as an improvised positioning jig.
Enabling attributes: molded bearing pockets, integrated ribs/posts, rigid skeleton, exposed/reachable
frame.
Affordance level: `Emergency 3`, meaning it is not the normal use, but is reasonably usable in a pinch.
Recipient condition: a cylindrical/rod-like item whose diameter is close enough to fit the pockets, rigid
enough to stay straight, and not too long/heavy.
Environment condition: adequate lighting is needed to visually align the rod with the pockets/posts.
Failure cases: rod too large or too small, rod too heavy/long, poor lighting, or moisture/poor fit causing
 slippage.

4. Gold Affordance Sampling

Affordance texts are embedded and clustered within each scenario. This groups affordances with related
functional roles. This example is sampled from:

Scenario: `home_office`
Affordance level: `3`
Cluster size range: `2-4`
Cluster ID: `290`

Thus, the sampled gold is not arbitrary: it is selected from a structured pool of scenario-specific,
clustered affordances.

5. Task Situation Construction

The task is generated from the gold affordance, but the prompt requires hiding the gold entity, part, and
mechanism. The recipient is chosen from the affordances recipient condition\/example recipients:

Recipient: a dry, straight wooden dowel, about 6 mm diameter and 12 cm long.

Constructed user-facing task:

The user is repairing a small wooden jewelry-box hinge made from the dowel. They need to keep the dowel
centered and steady while applying glue, but they do not have clamps or a proper jig.

This task is grounded because the dowel satisfies the KB recipient condition: it is cylindrical, rigid,
small enough, and compatible with the molded pockets.

6. Candidate Entity Sampling and Difficulty

The final task is marked:

Difficulty: `difficult`
Sampled entity count: `6`
```

```
The final scene includes the gold entity plus sampled candidate entities. In this example, the candidate
entities include:

`tape dispenser with weighted base`
`mobile filing cabinet with lock`
`clip-on ring light for webcam`
`keyboard tray with wrist rest`
`stainless steel insulated water bottle`
`bookshelf with adjustable shelves`
`digital desk timer with large display`

For difficult samples, the candidates are selected using comparison judgments, so the scene contains
plausible alternatives or distractors rather than random objects.

7. Final Solution

Prepare recipient: verify the dowel is dry, straight, close to 6 mm diameter, and short/light enough to
sit in the pockets.
Prepare use condition: expose the tape dispenser housing frame and place it flat so the molded pockets/
posts are reachable.
Prepare environment condition: position the desk lamp so the pockets and hinge area are clearly visible.
Apply affordance: seat the dowel in the molded bearing pockets/posts, align the jewelry-box hinge pieces
around the centered dowel, apply glue, and hold the pieces until the glue sets.
```

G Main Results Details

In this section, we present a more detailed and structured explanation of all our main results in Table 3. Additionally, we provide the statistical significance of main results in this appendix.

G.1 Main Results Details

Exact grounded tool use remains challenging. Although many models can moderately identify the correct entity, correctly grounding tool use at the part level remains challenging. Across models, entity correctness reaches 0.5149, while gold correctness drops to 0.1910, representing a relative decrease of over 60%. For some models, the gap reaches 0.35. Since our benchmark requires part-level attribute grounding to justify creative affordances, this gap suggests that models can often recognize which object might be useful but fail to identify the specific part that enables the intended affordance. Consequently, many predicted tool usages are plausible at a high level but lack the attribute-level grounding required for correct affordance discovery.

Reasonable actions do not imply physically grounded reasoning. Across models, the average score for action feasibility (3.5860) is noticeably higher than that for physical grounding (3.2003). This pattern indicates that models tend to propose actions that are intuitively plausible from commonsense reasoning, often relying on semantic alignment between the tool and the task, while the underlying justification based on physical attributes is incomplete or inaccurate. As a result, even seemingly reasonable actions may fail when evaluated against the detailed physical conditions required for successful execution, which is reflected in the lower prediction correctness scores (3.0327). A similar pattern appears in constraint coverage. Use constraints (C_u) and environmental constraints (C_e) are relatively well considered (3.1780 and 3.1860 on average), whereas recipient constraints (C_r) receive the lowest score (2.8026). This indicates that models frequently overlook conditions related to the target object or preparation steps needed before tool application, leading to incomplete reasoning pipelines even when the correct tool is selected.

Takeaway 1

From the metric perspective, current models are better at choosing plausible tools at the entity level but struggle to produce fully correct, physically grounded tool use. Model reasoning is often plausible but fails to account for all conditions required for successful execution.

Strong general models do not necessarily excel at creative tool use. We observe a notable contrast between general reasoning performance and creative tool-use ability. GPT-5.2 and GPT-5-Mini achieve the strongest performance on most reasoning-related metrics, including environmental constraint coverage, action feasibility, and overall prediction correctness. However, their gold correctness in tool usage (0.1819) is substantially lower than that of the Qwen3 family, particularly Qwen3-32B (0.2588) and Qwen3-14B (0.2483). In fact, Qwen3-32B achieves nearly 1.5× the gold correctness of GPT-5.2 while also obtaining the highest entity correctness (0.6246). This contrast suggests that some models are better at discovering unconventional yet valid tool usages, while others are stronger at evaluating constraints and producing grounded reasoning once the correct direction is identified. In other words, models that excel at systematic reasoning are not necessarily the ones that most effectively discover novel affordances.

Performance quickly saturates with model scaling. Within the same model family, scaling improves performance primarily at smaller model sizes but yields diminishing returns beyond a certain point. For example, in the Qwen family, gold correctness improves nearly 30% when scaling from 4B (0.1882) to 14B (0.2483), but increases by less than 5% when further scaling to 32B (0.2588). A similar trend appears in proprietary models: scaling GPT-5 Nano to GPT-5 Mini yields roughly a 40% relative improvement in gold correctness (0.1192 $\rightarrow$ 0.1687), whereas the gain from GPT-5 Mini to GPT-5.2 is only about 7%. Gemini models show a comparable pattern with modest gains of about 9%. These results indicate that creative tool use grounded in physical reasoning does not scale linearly with model size. Instead, it appears to depend on more fundamental capabilities related to affordance discovery and structured physical reasoning. This observation also highlights the value of our benchmark: the task cannot be easily solved through brute-force scaling alone and instead requires deeper improvements in creativity as well as grounded reasoning abilities.

Takeaway 2

From the model perspective, there is a clear split between creative affordance discovery and grounded reasoning quality. Improvements from model scaling alone quickly saturate, indicating that stronger performance requires advances in grounded affordance reasoning rather than size alone.

G.2 Statistical Significance of Main Results

To quantify the uncertainty associated with our main empirical findings, we conduct statistical tests for the headline model comparisons and the analyses reported in Sections 3, 5.2, and 5.3. We report mean effect sizes, confidence intervals, and p-values for all tested claims. Unless otherwise specified, comparisons are paired because the corresponding measurements are obtained from the same models under different metrics, task conditions, or inference settings. For ordered benchmark factors, such as gold cluster size and distractor similarity, we summarize the trend for each model across the ordered categories and test the resulting model-level trends.

For claims with a directional hypothesis, we use paired one-sided tests. All confidence intervals in **??** are 95% confidence intervals. We interpret a non-significant difference test as indicating that the data do not establish a reliable difference; it is not interpreted as evidence that the true effect is zero. For the preliminary Chain-of-Thought analysis, where we make a claim about the practical magnitude of the effect, we additionally conduct an equivalence test with an explicitly defined equivalence margin.

Physical grounding versus action feasibility. The main results show that models often generate actions that appear feasible while failing to ground those actions in the relevant physical properties. Across the 10 evaluated models, Physical Grounding is lower than Action Feasibility by 0.3657 points on average. The corresponding 95% confidence interval is $[-0.4851, -0.2462]$, and the paired one-sided test gives $p = 3.44 \times 10^{-5}$. This result provides strong statistical support for the claim that physical grounding is a more pronounced weakness than producing a superficially feasible action.

Effects of gold cluster size and distractor similarity. We next test two central observations from the benchmark-factor analysis. First, scores increase with gold cluster size. Across eight models, moving one step upward in the ordered gold cluster-size categories corresponds to a mean score increase of 0.0103, with a 95% confidence interval of $[0.0032, 0.0174]$ and $p = 0.00559$. This result supports the observation that affordances represented by larger clusters, and thus occurring more frequently in the annotated knowledge base, are easier for current models to recover.

Second, performance increases from dissimilar toward similar distractor settings. Across eight models, the mean trend is 0.0108, with a 95% confidence interval of $[0.0036, 0.0179]$ and $p = 4.62 \times 10^{-3}$. Although this result may initially appear counterintuitive, the fine-grained analysis in Figure 7 suggests that similar distractors can expose the relevant affordance dimension and guide models toward the physical properties that should be compared. The joint analysis of distractor count and similarity provides additional explanatory evidence for this result.

Effects of inference mode. Interactive evaluation performs significantly worse than static evaluation. Across the evaluated models, the mean Interactive-minus-Static difference is -0.1168, with a 95% confidence interval of $[-0.1524, -0.0811]$ and $p = 5.60 \times 10^{-5}$. The inspection statistics reported in the inference-mode analysis provide a behavioral explanation for this decline: models frequently commit to an answer before examining the gold entity and often terminate exploration after inspecting only a small subset of the available objects.

By contrast, structured CoT changes performance by only 0.0074 on average relative to the static setting. The 95% confidence interval is $[-0.0088, 0.0236]$, with $p = 0.317$. We therefore do not detect a statistically reliable improvement from CoT in the main CreativityBench evaluation. Because the confidence interval includes both zero and small positive effects, this difference test alone should not be interpreted as establishing that the two settings are equivalent.

Matched Qwen–GPT comparison. To evaluate whether general reasoning strength directly predicts performance on creative tool use, we compare Qwen3-32B/14B/4B with GPT-5.2/Mini/Nano at matched model tiers. Across the six paired comparisons formed from Gold Correctness and Entity Correctness, Qwen models score 0.0689 higher on average, with a 95% confidence interval of $[0.0063, 0.1314]$ and $p = 0.0183$.

The difference is particularly consistent for Gold Correctness. Qwen exceeds GPT at all three matched tiers, with a mean advantage of 0.0752, a 95% confidence interval of $[0.0615, 0.0889]$, and $p = 8.93 \times 10^{-4}$. These results support the observation that strong general-purpose reasoning performance does not necessarily translate into stronger part-level affordance discovery. This comparison should nevertheless be interpreted as evidence about the evaluated model families and tiers rather than as a universal ordering between open- and closed-source models.

Effects of model scaling. We compare each larger model with its next-smaller counterpart within the same family: GPT-5.2 versus GPT-5 Mini, Gemini-2.5-Pro versus Gemini-2.5-Flash, and Qwen3-32B versus Qwen3-14B. Across Gold Correctness and Entity Correctness, the average larger-model gain is 0.0115, with a 95% confidence interval of $[-0.0050, 0.0281]$ and $p = 0.0665$.

Thus, the evaluated comparisons do not provide statistically reliable evidence of an improvement from the larger variants on these objective metrics. However, the non-significant result is not evidence that scaling has no effect: the confidence interval remains compatible with small positive gains. We therefore interpret this analysis conservatively as showing that the observed upper-tier scaling gains are modest and statistically inconclusive in the evaluated model set.

Preliminary CoT difference tests. In the preliminary experiment from Section 3, structured CoT changes the mean score across the six evaluation dimensions by -0.0067 relative to

direct prompting. The 95% confidence interval is $[-0.1654, 0.1520]$, with $p = 0.541$. In the relative evaluation, CoT wins exceed direct-prompt wins by 1.67 percentage points on average, with a 95% confidence interval of $[-5.86, 9.20]$ percentage points and $p = 0.297$. These conventional difference tests do not detect a statistically reliable advantage for either prompting strategy. However, non-significance alone does not establish that the effect of CoT is practically small, motivating the equivalence analysis below.

Equivalence analysis of preliminary CoT results. We conduct a two one-sided equivalence test to directly assess whether the average CoT effect lies within a practically negligible range. For the absolute evaluation, we define the paired effect as CoT minus Direct and use an equivalence margin of ± 0.15 on the 1–5 score scale. We select this margin because a one-level improvement in only one of the six evaluation dimensions would change the average score by $1/6 \approx 0.167$; thus, 0.15 represents a conservative threshold for a minimally meaningful average improvement.

The mean effect is -0.0067, with a 90% confidence interval of $[-0.1311, 0.1177]$. This interval is fully contained within the equivalence range $[-0.15, 0.15]$, and the two one-sided equivalence test gives $p_{\text{TOST}} = 0.034$. This provides positive evidence that the average effect of CoT is practically bounded within the stated margin in the preliminary absolute evaluation.

For the relative evaluation, the CoT win-rate advantage is 1.67 percentage points, with a 90% confidence interval of $[-4.23, 7.57]$ percentage points. This interval is contained within the selected ± 10 percentage-point equivalence margin, where 10 percentage points correspond to fewer than one additional CoT win per ten comparisons. Together, these analyses support the narrow conclusion that CoT provides a practically limited average benefit under the explicitly defined margins. They do not imply that CoT has exactly zero effect or that it cannot improve particular evaluation dimensions or individual cases.

Interpretation. Taken together, the statistical analyses provide evidence for four main patterns. First, physical grounding is reliably weaker than action feasibility. Second, model performance varies systematically with gold cluster size and distractor similarity. Third, interactive evaluation reliably reduces performance, while the main-benchmark CoT comparison does not show a reliable improvement. Fourth, Qwen outperforms GPT in the evaluated matched-tier comparisons, whereas the observed gains from upper-tier scaling remain statistically inconclusive. We report effect sizes and confidence intervals alongside all p-values so that the magnitude and uncertainty of each finding can be assessed directly.

H Analysis Details

In this section, we present more details about our analysis settings and results, as a complementary for Section 5.3.

We analyze factors that influence model performance from three complementary perspectives. As described in Section 4, our task construction and sampling process is highly structured and balanced, enabling controlled analysis along several dimensions:

- **Gold Commonality:** How properties of the gold solution affect model performance, including affordance typicality level and the cluster size from which it is drawn.
- **Distraction Severity:** How distractors and environmental noise influence performance, focusing on both distractor counts and their affordance similarity to the gold solution.
- **Inference-Time Effects:** How inference-time choices, including sampling temperature and evaluation mode, affect model performance.

Unless otherwise specified, we analyze the **Tool Usage** metrics, which provide the most objective evaluation signal and serve as the primary benchmark metrics. The first two analyses use the same setting as the main experiment; for inference-time analysis, we further evaluate different sampling temperatures and evaluation modes. Finally, we examine subjective metrics and highlight several noteworthy patterns and failure modes.

H.1 How does Gold Commonality Affect Performance?

To understand how the inherent commonality of the gold solution influences model performance, we analyze tasks along two dimensions. First, we group tasks by the **size of the gold cluster** from which the solution is sampled, using three ranges: [2, 4], [5, 10], and [10, 50]. A larger cluster generally indicates that the underlying affordance appears more frequently in the annotated data and is therefore more common, even if it is still unconventional relative to the object's default use. Second, we analyze the **emergency level** of the gold solution, which captures how natural or practically acceptable an affordance is in real-world scenarios; as such, it also serves as a proxy for the commonality and naturalness of the sampled affordance.

More common affordances are consistently easier for current models. We report the results in Figure 5. For gold cluster sizes, performance increases steadily with gold cluster size (Figure 5a). Across models, tasks drawn from small clusters (size 2–4) produce the lowest scores, while those drawn from larger clusters are significantly easier. The same pattern appears when grouping by emergency level (Figure 5b). Across nearly all models, affordances at levels 1–2 are associated with noticeably lower performance than those at levels 3–5. This result reinforces the same conclusion from a different angle: current models handle relatively common repurposed affordances much better than rare or highly atypical ones.

Together, these results show that model performance is strongly shaped by how familiar or natural the target affordance is. This is consistent with the broader intuition that current models remain weak on long-tail or unusual uses of objects, even when they can perform reasonably well on more common forms of creative repurposing. At the same time, the consistency of this trend also provides indirect support for the validity of our annotation design: the commonality-related labels appear to align well with both human intuition and observed model difficulty.

Takeaway 3

Model performance drops most clearly on rare, long-tail affordances, showing that commonality remains a major driver of success in grounded creative tool use.

H.2 How does Distraction Severity Affect Performance?

Beyond the presence of the gold entity itself, distractor objects in the environment introduce additional cognitive load for the tested models. We analyze this effect from two complementary perspectives. First, we examine the **number of distractors** presented in each task. We group tasks by the total number of distractor entities (3, 6, 9, and 12). A larger set of entities increases the length of the context and expands the space of candidate tools the model must evaluate. Second, we analyze the **affordance similarity** between distractors and the gold object. During dataset construction (Section 4), distractors were categorized based on whether they could elicit attributes similar to the gold affordance during verification. We therefore divide distractors into two categories: *similar* distractors, which share related affordance attributes with the gold object but are ultimately inferior solutions, and *dissimilar* distractors, which do not naturally support the target affordance. Based on these labels, we group tasks into three settings: all-similar distractors, all-dissimilar distractors, and mixed distractors.

More distractor entities increase reasoning difficulty. We report the results in Figure 6. The left panel shows performance as the number of distractors increases. Across all tested models, performance consistently decreases as more entities are introduced. On average, the drop is more noticeable when moving from 3 to 9 distractors, while the decline from 9 to 12 becomes somewhat less steep, suggesting a potential diminishing effect once the candidate set becomes large. Overall, the results indicate that increasing the number of entities makes it harder for models to identify the correct tool, likely because the model must allocate attention across more competing options.

Similar affordances can implicitly guide model reasoning. The right panel of Figure 6 examines the role of distractor similarity. Intuitively, one might expect distractors with similar affordances to increase task difficulty, since they require finer-grained comparison between candidate objects. However, the results reveal the opposite pattern: tasks containing similar-affordance distractors consistently achieve higher scores than those containing dissimilar distractors. A possible explanation is that similar distractors implicitly highlight the relevant affordance space. When multiple entities share related attributes, the intended affordance becomes more salient, making it easier for the model to activate the correct reasoning pathway.

To further investigate this effect, we perform a fine-grained analysis of distractor number within the similar and dissimilar groups separately. As shown in Figure 7, for tasks with dissimilar distractors, performance steadily decreases as the number of entities grows, consistent with the expected effect of increased distraction. In contrast, for tasks with similar distractors, the trend is less monotonic: performance initially decreases but then partially recovers when the number of entities reaches 9. This pattern suggests that while more entities increase the overall reasoning burden, the presence of multiple affordance-related objects may also reinforce the relevant affordance concept and help guide the model toward the correct solution.

Takeaway 4

Both the quantity and type of distractors influence model performance. While larger candidate sets consistently increase difficulty, distractors that share similar affordances with the gold object can sometimes help highlight the relevant reasoning space, partially offsetting the distraction effect.

H.3 How do Inference Settings Affect Performance?

Beyond model capability, inference-time strategies may also influence performance. We analyze two aspects of inference configuration: the **sampling strategy** and the **evaluation mode**. First, we examine whether increasing sampling temperature encourages more creative reasoning. While the main experiments use deterministic decoding, we additionally evaluate models with higher temperatures ($T = 0.7$ and $T = 1.0$) to test whether more diverse sampling leads to better tool discovery. Second, we explore alternative evaluation modes. In the main setting, all entity descriptions are presented at once in a static prompt. We compare this with two variants:

- **CoT mode.** The model is instructed to explicitly perform a structured reasoning process, including task analysis, attribute grounding, and affordance reasoning, before predicting the entity and usage strategy.
- **Interactive mode.** Instead of receiving all entity descriptions upfront, the model need to query the environment and request entity descriptions one at a time, turning the task into a multi-turn interaction.

We sample 10% of the original data (1.4K) for the analysis of this section, and run multiple parallel experiments for comparison. All other settings remain identical to the main experiments. Please see Section H for more details.

Higher temperature does not guarantee genuine creative reasoning. We report the results in Figure 8. The left panel show that increasing temperature generally degrades performance for smaller-scale models like Qwen series, and mildly increase performance for larger-scale models like GPT-5.2. Inspection of model outputs reveals that higher temperatures primarily increase hallucination rather than useful exploration. As temperature rises, smaller models are more likely to generate entity or part names that do not exist in the environment. This suggests that creative tool use in our benchmark is fundamentally different from open-ended text generation: successful solutions require grounded reasoning over attributes and affordances, rather than simply producing diverse outputs.

These results also imply that naive inference scaling such as repeatedly sampling at higher temperatures may not effectively and consistently improve performance across models.

Increased diversity alone does not help models discover the correct affordance relationships and instead may lead to ungrounded answers.

Takeaway 5

Increasing sampling temperature does not reliably improve creative tool use. For smaller models in particular, higher temperatures tend to increase hallucinated entities and parts rather than produce more effective affordance exploration, suggesting that this benchmark rewards grounded constraint satisfaction rather than open-ended generative diversity.

Interactivity introduces additional reasoning challenges. The right panel of Figure 8 compares the static setting with the interactive and structured-CoT inference modes. Interactive evaluation substantially reduces performance across all models. Although this setting lowers the initial cognitive load (since entity descriptions are revealed only upon request rather than all at once), models exhibit limited exploration behavior. According to Section 5.3, most models inspect fewer than three entities on average before producing a final prediction.

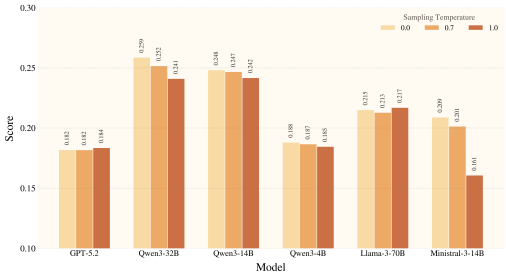
This limited exploration makes early mistakes especially costly. Once a model forms an initial hypothesis about which object may solve the task, it seldom gathers enough additional evidence to revise that belief. The gold inspection statistics in **??** further support this pattern. When both the predicted entity and part are correct, the gold inspection rate is high, indicating that successful predictions are usually grounded in direct examination of the target object. In contrast, for completely incorrect predictions, the gold inspection rate falls below 20% for most models, showing that the model often never inspects the gold entity before answering. Interestingly, even among fully correct cases, the gold inspection rate does not reach 100%, suggesting that models sometimes succeed based on partial clues or rough heuristics rather than thorough exploration. Overall, these results indicate that insufficient exploration and premature commitment to early hypotheses are major sources of failure in the interactive setting.

Structured CoT introduces only minor performance changes. In contrast to the large performance drop in interactive mode, structured CoT reasoning results in only modest changes. Some models, especially those in the Qwen family, obtain small gains (typically within around 5%), while GPT-family models even show slight declines. This observation is consistent with our earlier findings in Section 3: explicitly enforcing a structured reasoning process does not substantially improve affordance reasoning.

A possible explanation is that structured CoT changes the format of reasoning more than its substance. Once a model commits to a particular entity early in the reasoning process, the imposed structure may further reinforce that focus rather than encourage broader comparison across alternatives. In practice, models rarely revisit previously overlooked entities or perform the kind of back-and-forth comparison needed to identify the best tool. As a result, structured reasoning alone offers limited benefit when the core challenge is to explore the candidate space and ground the final decision in comparative affordance analysis.

Takeaway 6

Changing the inference mode also yields limited gains. Interactive evaluation reveals weak exploration behavior and premature commitment to early hypotheses, while structured CoT provides only marginal improvement, indicating that the main bottleneck is not reasoning format but the ability to compare candidates and ground decisions in affordance-level physical reasoning.



(a) Inference temperature

Model	Static	Interactive	CoT
GPT-5-mini	0.169	0.088 −0.081	0.162 −0.006
GPT-5-nano	0.119	0.056 −0.064	0.116 −0.003
Qwen3-32B	0.259	0.093 −0.166	0.261 +0.002
Qwen3-14B	0.248	0.099 −0.150	0.260 +0.012
Qwen3-4B	0.188	0.062 −0.127	0.201 +0.013
Llama-3-70B	0.215	0.047 −0.168	0.250 +0.035
Ministral-3-14B	0.209	0.081 −0.128	0.247 +0.038

(b) Inference mode

Figure 12: Additional results on full test set evaluation on different inference-time sampling temperatures and evaluation mode applied.

H.4 Inference setting's impact on performance

For Section H.3 in the main text, we evaluate only 10% of the current dataset. We make this choice for two reasons. First, this subset still contains about 1.4K examples, which is already large enough to support reliable analysis. Second, the required experimental budget is substantial: running the full 14K-example set across all models is expensive, and the cost becomes even higher for interactive-mode evaluation because it requires multi-turn inference. In addition, for the temperature-sampling experiments, we exclude GPT-5 Mini and GPT-5 Nano because their sampling temperature cannot be controlled.

Nevertheless, to validate that the sampled subset is representative, we also evaluate the full 14K test set on selected models and report the effects of sampling temperature and inference mode in Figure 12. The full-set results show trends consistent with those reported in the main text. For temperature sampling, the performance of GPT-5.2 and Llama-3-70B tends to improve, whereas smaller models generally experience performance drops. A similar pattern appears in the inference-mode experiments: interactive mode substantially reduces performance, while CoT mode leads only to small fluctuations and yields no clear overall benefit.

Overall, these full-set results provide additional evidence that the 10% sampled subset is representative of the broader dataset.

H.5 How are auxiliary metrics affected?

In this section, instead of focusing on the main Tool Usage metrics, we analyze auxiliary metrics evaluated by the LLM judge. All analyses follow the same setup as the main table. Since we consider multiple factors, including gold cluster size, gold emergency level, distractor similarity, and distractor number, together with six auxiliary scores, we only discuss several representative patterns here and leave the full results to the next subsection.

Physical grounding is harder for rare affordances and dissimilar distractors. We report four representative results in Figure 13. From panels (a) and (b), we observe that across all tested models, gold affordances from smaller clusters consistently achieve the lowest performance. This trend also remains when distractors are all dissimilar, and is consistent with the main Tool Usage analysis results in the previous section.

This suggests that rare affordances are harder for models to ground physically in creative tool use. Since such affordances are less common, models may struggle to reason concretely about how the object can be used. Similarly, dissimilar distractors offer little affordance-level guidance, making it harder for models to identify and ground the effective use of the target part.

Auxiliary metrics further show that similar distractors make the task easier. From panels (b) and (d), we see that similar-affordance distractors lead to higher physical grounding

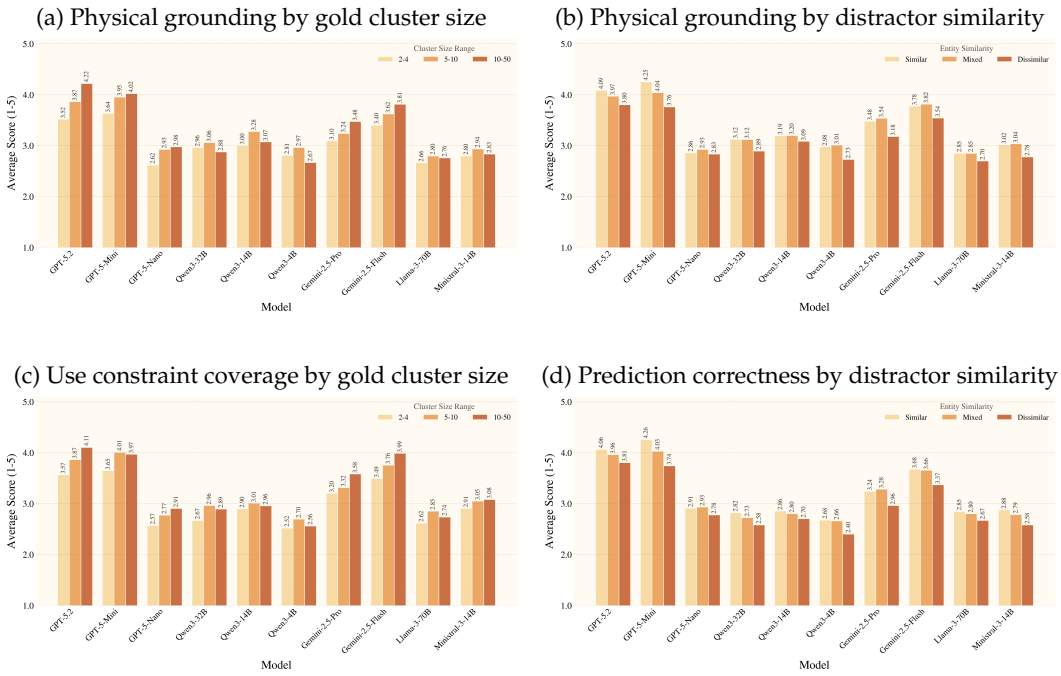


Figure 13: Fine-grained statistics showing how auxiliary metrics are affected from multiple perspectives.

scores and higher prediction correctness. Although such distractors increase surface-level ambiguity, they also constrain the solution space to plausible uses, making the task easier for models.

Rare affordances are also harder to use under appropriate conditions. Panel (c) shows that models attend more to use conditions when the gold affordance comes from a larger cluster, i.e., a more common creative affordance. This suggests that rare affordances require more precise conditional reasoning, which current models often fail to capture. As a result, models may either ignore these conditions or reason about them incorrectly.

Takeaway 7

Auxiliary metrics reveal a consistent pattern: models handle familiar affordances much better than rare ones, and they benefit from distractors that provide affordance-level cues. Overall, current models still struggle to ground rare, weakly signaled affordances in a physically and conditionally appropriate way.

H.6 Fine-grained analysis on auxiliary metrics

From Figure 14 to Figure 17, we present the complete results of the fine-grained analysis of the auxiliary metrics on different analysis aspects including gold affordance cluster size, gold affordance emergency level, distractor similarity, and distractor number.

H.7 Error analysis details

For error analysis, we use Gemini-3.1-Flash-Lite as the LLM judge on a 10% sample of failure cases for each model. The temperature is set to zero to ensure deterministic judgments. The judging protocol largely follows the same criteria used in the main-results analysis.

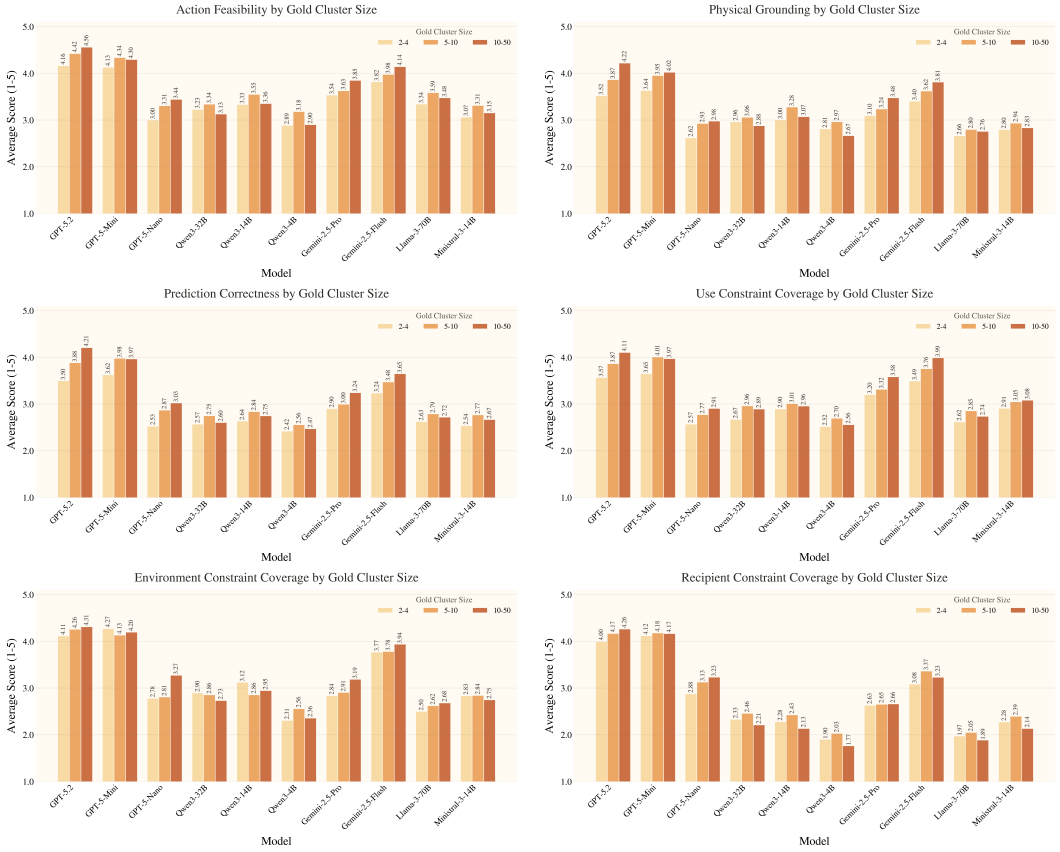


Figure 14: Fine-grained auxiliary metrics analysis upon different gold affordance cluster sizes.

More specifically, the judge is instructed to evaluate each case from two perspectives. First, it assesses the prediction on its own, focusing on constraint satisfaction, physical grounding, and action feasibility. Second, it compares the prediction against the gold solution, using the gold supporting rationale introduced during task creation in Section 4. In this comparative step, the judge determines whether the gold solution is indeed preferable, or whether the prediction still has meaningful merits, and summarizes the respective pros and cons of both.

Note that, consistent with the analysis in the main experimental setting and the preliminary analysis in Section 3, we rescale the judge's original score from the range 0–2 to 1–5 after scoring. The detailed prompts for these two judgment aspects are provided below.

System Prompt for Fine-grained Metric-wise Judgment

```
You are a strict evaluator for practical goal completion.
The predicted entity/part is NOT exactly gold.
Judge whether the predicted "how_to_use" can still accomplish the task reasonably and safely.

Task:
{task.get("task", "")}

Gold solution JSON (only for reference):
{json.dumps(task.get("solution", {}), ensure_ascii=False, indent=2)}

Physical attributes of the predicted part:
{json.dumps(pred_physical_attrs, ensure_ascii=False, indent=2)}

State attributes of the predicted part:
{json.dumps(pred_state_attrs, ensure_ascii=False, indent=2)}
```

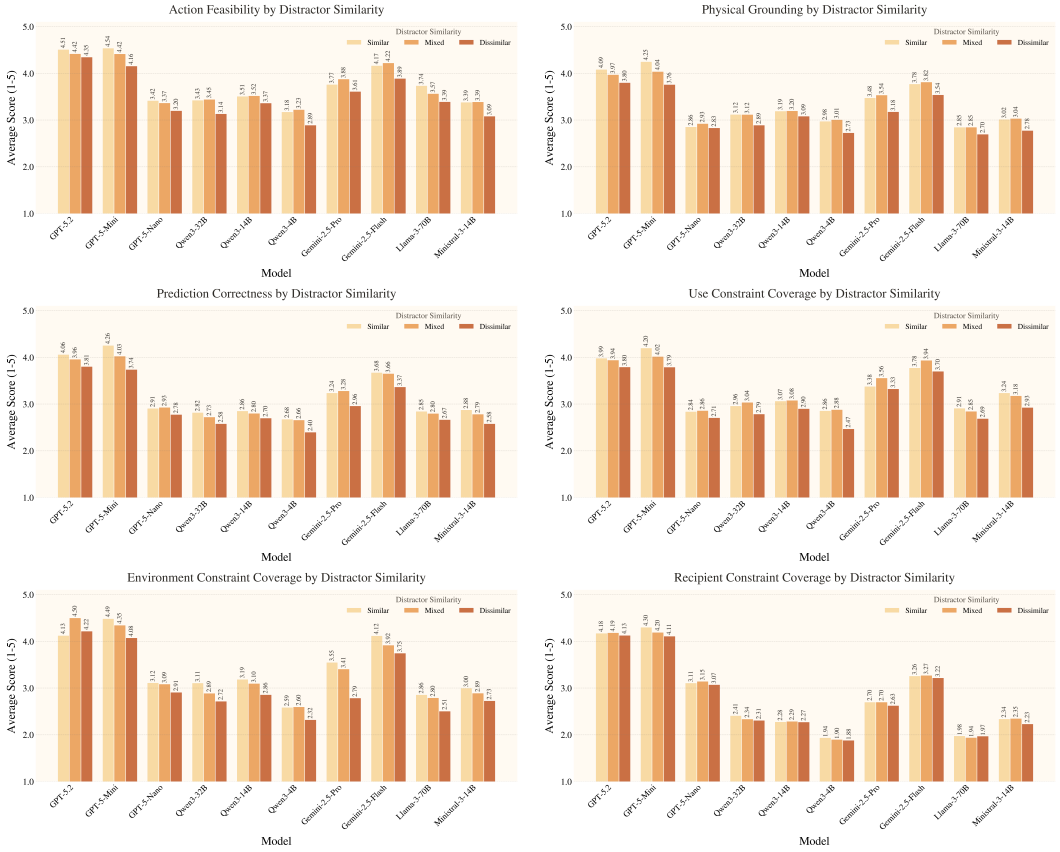


Figure 15: Fine-grained auxiliary metrics analysis upon different distractor similarity tiers.

```
Predicted entity to use: {pred_entity}
Predicted part to use: {pred_part}
Predicted how to use the part:
{how_to_use}

Field-by-field rubric when judging the predicted "how_to_use":
1) environment_condition_covered (0/1/2/"NA"): [Do NOT directly compare with the gold, it's only for
reference]
- "NA": in your reasoning, deeply consider for the predicted use of the part to accomplish the task, if
the external environment setup is needed; if not, then please give "NA".
- 0: the environment setup is needed but in the predicted use, it is not mentioned or covered at all.
- 1: the environment setup is needed but in the predicted use, it is mentioned but not all are covered.
- 2: the environment setup is needed but in the predicted use, it is covered well and reasonable, aligning
 with common sense.

2) use_condition_covered (0/1/2/"NA"): [Do NOT directly compare with the gold, it's only for reference]
- "NA": in your reasoning,deeply consider for the predicted use of the part to accomplish the task, if the
 preparation/access of the part is needed; if not, then please give "NA".
- 0: the preparation/access of the part is needed but in the predicted use, it is not mentioned or covered
 at all.
- 1: the preparation/access of the part is needed but in the predicted use, it is mentioned but not all
are covered.
- 2: the preparation/access of the part is needed but in the predicted use, it is covered well and
reasonable, aligning with common sense.

3) recipient_condition_covered (0/1/2/"NA"): [Do NOT directly compare with the gold, it's only for
reference]
- "NA": in your reasoning, deeply consider for the predicted use of the part to accomplish the task, if
the recipient-side prerequisites are needed; if not, then please give "NA".
- 0: the recipient-side prerequisites are needed but in the predicted use, it is not mentioned or covered
at all.
- 1: the recipient-side prerequisites are needed but in the predicted use, it is mentioned but not all are
 covered.
```

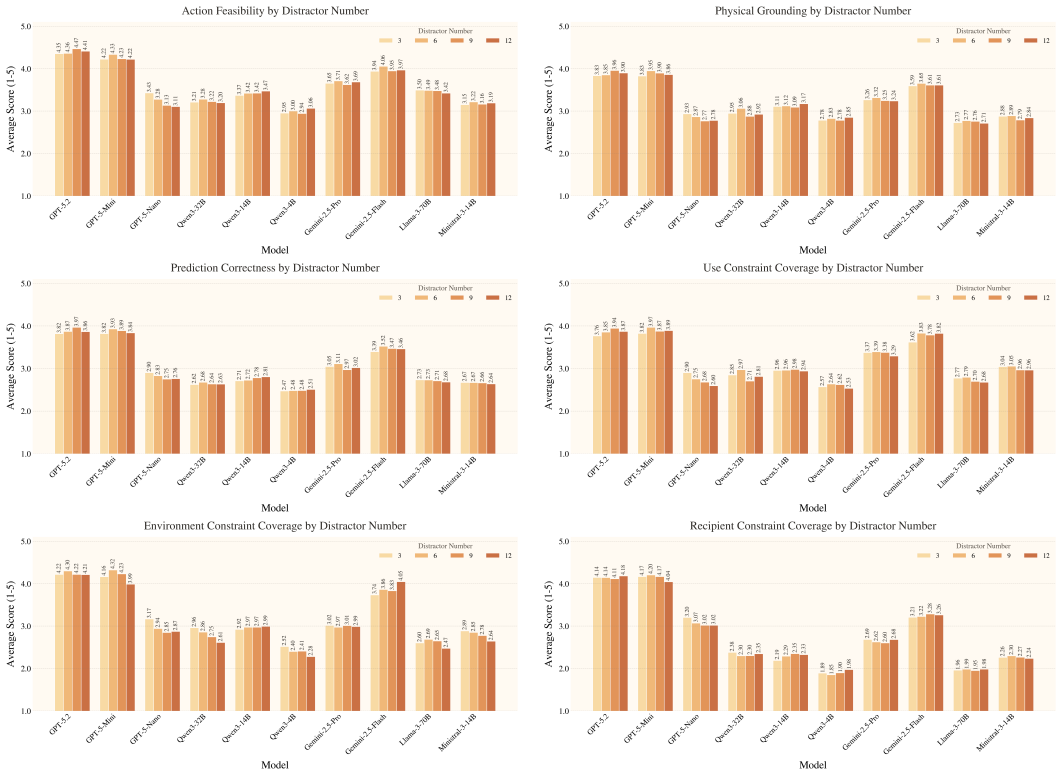


Figure 16: Fine-grained auxiliary metrics analysis upon different distractor numbers.

```
- 2: the recipient-side prerequisites are needed but in the predicted use, it is covered well and
reasonable, aligning with common sense.

4) attributes_grounding (0/1/2): [Compare with the physical and state attributes of the predicted part]
- 0: the predicted action is not grounded in the key enabling attributes of the part, or it violates some
attributes of the part.
- 1: the predicted action is mostly grounded in the key enabling attributes of the gold affordance, but
not all are covered or implied.
- 2: the predicted action is fully grounded in the key enabling attributes of the gold affordance, and
most of them are explicitly mentioned, covered or implied.

5) action_feasibility (0/1/2): [Please only focus on the predicted action itself]
- 0: the action itself is physically impossible, not working at all, very unlikely to be used in practice,
 or unsafe.
- 1: the action itself is partially workable but still there are some steps that are not plausible,
aligned with common sense or not feasible.
- 2: the action itself is operationally correct and complete, almost completely aligned with common sense
and feasible in practice.

Evidence policy:
- Every *_reason must quote concrete evidence from predicted how to use the part and other given relevant
information.
- If evidence is unclear/missing, default to a stricter outcome (lower score).
- Make your reasoning clear, concise, and to the point for each field, and your score should be based on
the evidence.

Please make sure to only return a valid JSON with exactly these 12 fields (no extras, no markdown).
{{
    "environment_condition_covered_reason": "...",
    "environment_condition_covered": 0/1/2/"NA",
    "use_condition_covered_reason": "...",
    "use_condition_covered": 0/1/2/"NA",
    "recipient_condition_covered_reason": "...",
    "recipient_condition_covered": 0/1/2/"NA",
    "attributes_grounding_reason": "...",
    "attributes_grounding": 0/1/2,
    "action_feasibility_reason": "...",
    "action_feasibility": 0/1/2,
```

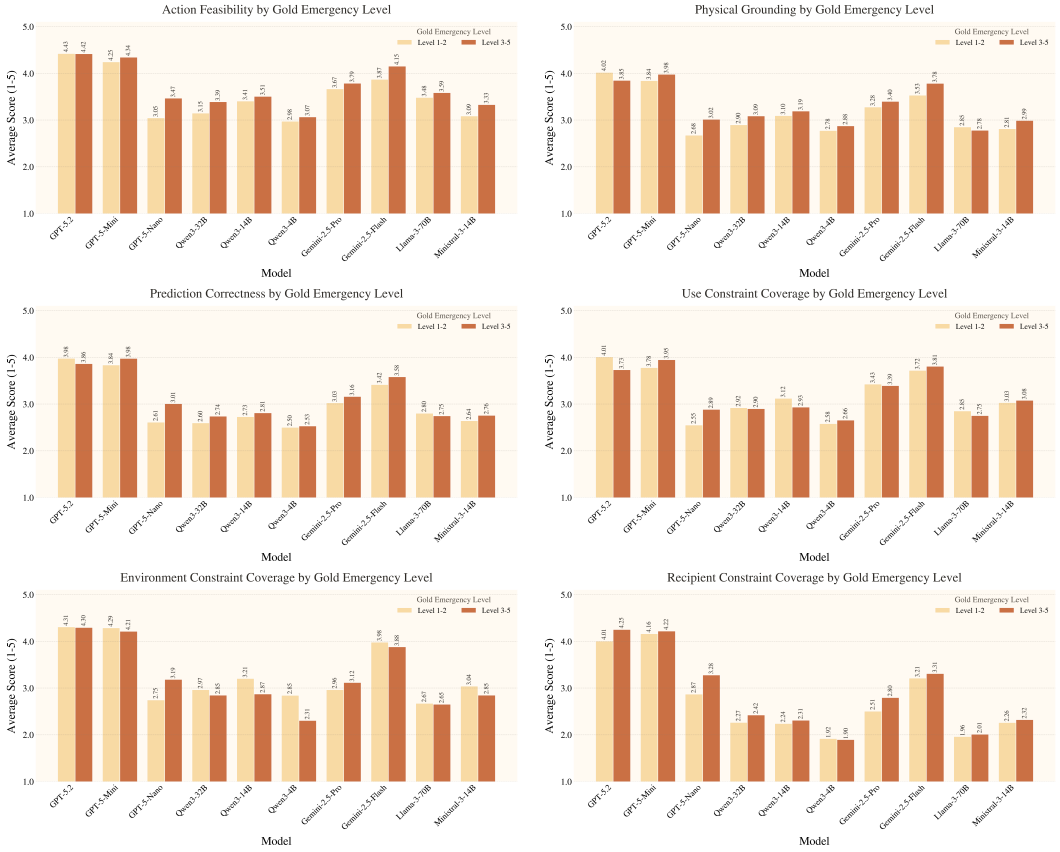


Figure 17: Fine-grained auxiliary metrics analysis upon two grouped gold emergency level bands.

```
}}
```

System Prompt for Gold-Prediction Comparison Judgment

```
You are a strict evaluator comparing predicted substitution vs ground-truth preference.
You are given the model prediction and the a ground-truth comparison judgment for that predicted entity+
part.
Judge how convincing the model prediction remains AFTER reading the comparison evidence.

Task:
{task.get("task", "")}

Predicted entity to use: {pred_entity}
Predicted part to use: {pred_part}
Predicted how to use the part:
{how_to_use}

Ground-truth comparison judgment for the predicted entity+part:
{json.dumps(pred_comp if pred_comp else {}, ensure_ascii=False, indent=2)}

Return ONLY JSON:
{{
  "prediction_reasonable_reason": "...",
  "prediction_reasonable_level": 0 or 1 or 2
}}

Scoring rubric:
- 0: prediction is somewhat reasonable, but ground truth is clearly better after reading the comparison
judgment.
```

Model	Percentage	Gold Comparison	Constraint Coverage			Physical Grounding	Action Feasibility
			Use (C_u)	Env. (C_e)	Rcpt. (C_r)		
Closed-Source Models							
GPT-5.2	47.90%	1.2889	4.5297	4.4617	4.6226	4.3874	4.2980
GPT-5 Mini	51.41%	**1.2942**	**4.7010**	**4.5982**	**4.7112**	**4.5513**	**4.4560**
GPT-5 Nano	42.79%	1.1489	3.9395	4.3547	4.3805	3.4507	3.2759
Gemini-2.5-Pro	64.48%	1.1651	4.3639	3.0100	3.2293	4.3829	3.6117
Gemini-2.5-Flash	63.06%	1.1854	4.5111	3.5451	3.6860	4.3584	3.6899
Open-Source Models							
Qwen3-32B	37.47%	1.1443	3.4811	2.7159	2.6444	3.5012	2.6879
Qwen3-14B	38.70%	1.1252	3.0770	2.5366	2.3411	3.1419	2.3920
Qwen3-4B	47.26%	1.0871	2.4423	2.0862	1.8587	2.6029	1.8733
Llama-3-70B	44.67%	1.1329	2.4609	2.1490	2.1086	2.5749	2.1575
Ministral-3-14B	47.37%	1.0988	3.4519	2.7042	2.6591	3.0195	2.3329
Average	48.51%	1.1671	3.6958	3.2162	3.2241	3.5971	3.0775

Table 13: Error analysis summary on judged wrong outputs where **both the entity and the part are wrong**. Percentage is the share of this slice over all tasks for that model. Best and second-best results in each score metric are highlighted in **bold** and underline, respectively. The maximum score for each non-percentage metric is 5.0. Higher is better for all metrics.

Model	Percentage	Gold Comparison	Constraint Coverage			Physical Grounding	Action Feasibility
			Use (C_u)	Env. (C_e)	Rcpt. (C_r)		
Closed-Source Models							
GPT-5.2	33.90%	1.3495	3.8951	4.4197	4.5242	3.3727	3.7184
GPT-5 Mini	30.97%	1.3433	**4.2242**	**4.5805**	**4.6413**	3.6712	**3.9150**
GPT-5 Nano	41.86%	1.1763	3.2356	4.1729	4.3309	2.4372	2.9408
Gemini-2.5-Pro	18.82%	**1.3765**	4.1961	3.0943	3.1685	**4.0647**	3.6064
Gemini-2.5-Flash	21.62%	1.3246	4.0685	3.5596	3.5083	3.5011	3.4336
Open-Source Models							
Qwen3-32B	35.34%	1.2553	3.2526	2.7622	2.5685	3.3099	2.7903
Qwen3-14B	36.56%	1.2252	2.9267	2.5558	2.4404	2.8104	2.5426
Qwen3-4B	33.94%	1.2055	2.5440	2.2144	2.0052	2.5506	2.1610
Llama-3-70B	33.82%	1.2290	2.5096	2.2587	2.2252	2.2489	2.3660
Ministral-3-14B	30.92%	1.2179	3.3285	2.7540	2.6790	2.7024	2.4532
Average	31.77%	1.2703	3.4181	3.2372	3.2091	3.0669	2.9927

Table 14: Error analysis summary on judged wrong outputs where **the entity is correct but the part is wrong**. Percentage is the share of this slice over all tasks for that model. Best and second-best results in each score metric are highlighted in **bold** and underline, respectively. The maximum score for each non-percentage metric is 5.0. Higher is better for all metrics.

```
- 1: prediction has meaningful merit; trade-offs are close and prediction may be acceptable after reading
the comparison judgment.
- 2: prediction is strongly convincing and should replace current gold choice after reading the comparison
 judgment.

Reasoning requirements:
- Reference available comparison aspects (accessibility, side-effects, willingness, commonness, safety,
etc.).
- Consider whether predicted "how to use the part" strengthens or weakens the substitution.
- Penalize unsupported claims.

Please make sure to only return a valid JSON with exactly these 2 fields (no extras, no markdown).
```

H.8 Error analysis details

In this section, we analyze the cases where the model selects an incorrect entity or part as the tool, in order to understand how models behave when tool selection fails. We divide the errors into two categories: (1) cases where the entity is correct but the part is wrong, and (2)

cases where both the entity and the part are wrong. The results are reported in Table 14 and Table 13, respectively.

Our analysis focuses on two aspects. First, we compare the predicted solution with the gold solution. For each task, we ask a third-party LLM to judge which solution is more convincing on a scale of 1–5, where 1 indicates the gold solution is much more convincing and 5 indicates the prediction is much more convincing. During this comparison, we also provide the gold justification annotations collected during the affordance verification step described in Section 4.

Second, we evaluate the predicted solutions themselves, even though they are associated with incorrect tools. Specifically, we assess their **constraint coverage**, **physical grounding**, and **action feasibility**, using the same criteria as in the main evaluation. We do not evaluate prediction correctness here because the prediction is intentionally conditioned on an incorrect tool, making correctness with respect to the gold solution less meaningful.

It is important to note that not all metrics are directly comparable with those in Table 3. The gold comparison score measures relative preference between two solutions, while the other metrics assess the intrinsic quality of the predicted solution. Moreover, when a model uses a different tool from the gold one, the predicted procedure may follow a different reasoning path, which further limits direct comparison with the main table.

All judgments are performed using Gemini-3.1-Flash-Lite. We adopt this automated evaluation because the number of erroneous cases is large (nearly 120K). Through human validation and sampling checks, we confirm that the model provides sufficiently reliable and cost-effective judgments.

Gold solutions are overwhelmingly preferred. Across both error categories, the gold solutions dominate the comparison. The average gold comparison scores are close to 1, corresponding to a gold win rate exceeding 95%. This confirms that when the model selects an incorrect tool, the resulting solution is rarely competitive with the correct one.

We also observe a clear difference between the two error types. When both the entity and the part are wrong, the predicted solutions are judged slightly worse than when only the part is wrong. This is intuitive: using the correct entity still preserves some semantic alignment with the task, while selecting a completely wrong tool often leads to more obviously implausible reasoning.

Open-source models degrade more severely under tool errors. Even when evaluated independently of the gold solution, open-source models consistently receive lower scores in constraint coverage and physical grounding compared to closed-source models. This indicates that their generated procedures themselves are of lower quality once the tool choice deviates from the correct one.

When compared with Table 3, this gap becomes more apparent. For open-source models, incorrect tool selection leads to a noticeable drop in reasoning quality. In contrast, strong closed-source models such as GPT-family models often still produce internally consistent explanations even when the chosen tool is not optimal. Although these predictions are still inferior to the gold solutions, they remain relatively well-grounded and coherent.

Action feasibility drops the most under tool errors. Among all metrics, action feasibility shows the largest degradation. Unlike others, this metric is directly comparable with the results in Table 3 because it only evaluates whether the proposed action itself is plausible, regardless of the gold solution.

We observe an average absolute decrease of nearly 0.6. This suggests that once the model selects the wrong tool, it becomes substantially harder to construct a plausible action sequence. In many cases, the generated procedure no longer aligns with common-sense physical interactions, making the proposed actions appear unnatural or infeasible.

Takeaway 8

Correct tool selection is critical not only for final correctness but also for maintaining coherent and physically plausible reasoning. When the wrong tool is used, solution quality deteriorates sharply (especially in action feasibility), and the degradation is substantially more severe for open-source models.

H.9 Attribution analysis details

In the previous section, we showed that the gold solution is strongly preferred by the judge, even in cases where the predicted "how to use" is itself plausible despite not matching the gold tool. In this section, we further investigate why this preference arises through an attribution analysis. Specifically, for each model, we randomly sample 10% of its failure cases and use Gemini-3.1-Flash-Lite as the categorization model to identify the main reasons the prediction is considered inferior to the gold solution.

We categorize the reasons why the gold solution is preferred (or why the prediction fails) into four high-level groups that capture different types of failure modes in tool repurposing. Each group further contains several fine-grained categories that describe specific mechanisms of failure.

- **A. Physical invalidity.** The proposed repurposed tool cannot realistically perform the task due to incorrect physical assumptions. This includes: (i) *A1. Hallucinated affordance* (the model assumes non-existent features), (ii) *A2. Affordance mismatch* (the object's geometry, material, or mechanics are unsuitable), and (iii) *A3. Performance shortfall* (the object lacks sufficient capacity such as stability, mass, or precision).
- **B. Practical infeasibility.** The solution is impractical to execute in real-world settings. This includes: (i) *B1. Destructive workaround* (requiring dismantling or damaging objects), and (ii) *B2. Context or accessibility issues* (the object is hard to access, or the procedure is overly complex).
- **C. Risk or constraint mismatch.** The proposal introduces risks or violates task constraints. This includes: (i) *C1. Safety or damage risk* (unsafe, unhygienic, or likely to cause damage), and (ii) *C2. Constraint violation* (contradicting explicit requirements or intended object use).
- **D. Comparative inferiority.** The prediction is workable but still worse than the gold solution. This includes: (i) *D1. Inferior but workable solutions* (less stable, convenient, or robust), and (ii) *D2. Preference-sensitive comparisons* (both solutions are reasonable, but the gold is more standard or practical).

This taxonomy allows us to distinguish between fundamentally incorrect physical reasoning, impractical procedures, violations of real-world constraints, and cases where the prediction is merely less preferable rather than strictly incorrect. For each case, we assign one primary category and optionally several contributing categories, and report their distributions respectively in the following analysis.

Physical invalidity is the dominant source of failure. Across models, physical invalidity (Category A) is by far the most common reason the gold solution is preferred, both as the primary cause and as a contributing factor. Within this group, A2 (*affordance mismatch*) is especially prominent, showing that models often choose objects whose geometry, material, or mechanics do not actually support the intended function. This suggests that the core weakness is not simply poor planning, but insufficient grounding in the physical properties of candidate tools.

Many errors reflect over-attribution of affordances. A1 (*hallucinated affordance*) is also highly frequent, indicating that models sometimes assign capabilities to objects that they do not realistically have. These cases suggest a tendency to optimize for the task goal at the level of abstract function while under-checking whether the selected object can truly realize that function. As a result, the predicted solution may sound plausible in isolation, yet still fail because it relies on imagined or overstretched affordances.

Practicality and safety remain important secondary factors. Although less dominant than physical invalidity, practical infeasibility (B) and risk or constraint mismatch (C) also appear in a substantial share of contributing reasons. This means that even when a solution is not fundamentally impossible, it may still be inconvenient, unsafe, or inconsistent with task constraints, which makes it less preferable than the gold solution. By contrast, comparative inferiority (D) is relatively less common as a contributing cause, suggesting that most failures are not merely weaker alternatives, but are rooted in deeper issues of physical and real-world grounding. Please see more model-wise analysis and details in Section H.

Takeaway 9

Most failures come from weak physical grounding: models often over-imagine what objects can do or choose tools whose real properties do not support the intended use. Practicality and safety matter too, but far fewer errors are simply suboptimal alternatives rather than genuinely mismatched solutions.

H.10 Fine-grained analysis on attribution analysis

We further conduct the attribution analysis on a randomly sampled 10% subset of cases using Gemini-3.1-Flash-Lite as the categorization model, with the judging temperature fixed at 0.0. To support the reliability of using Gemini as the judge, we additionally repeat the same categorization on the same sampled subset with Qwen3-32B and GPT-5-Mini. We find that Gemini's predicted primary category overlaps with Qwen3-32B in 83.3% of cases and with GPT-5-Mini in 90.91% of cases. This high agreement suggests strong cross-model consistency in the attribution judgments, which supports the validity of the taxonomy assignment and justifies our use of Gemini-3.1-Flash-Lite as the main categorization model.

Specifically, we feed the gold comparison reason into the categorization judge and instruct it to assign exactly one primary contributing factor, along with any additional contributing categories when appropriate. The detailed prompt is shown below:

System Prompt for Gold-Prediction Comparison Judgment

```
You are an expert judge for error analysis of failed physical tool-use predictions.

You will receive one sentence that explains why a prediction was judged unreasonable.
Your job is to classify the failure reason into the taxonomy below.

Categories:

A. Physical / functional invalidity
A1. Hallucinated affordance / wrong object model
- The prediction invents a feature, subpart, or capability the object does not actually have.

A2. Affordance mismatch (wrong geometry / material / mechanics)
- The object is real, but its actual shape, material, or operating principle is fundamentally wrong for
the task.

A3. Performance mismatch (too weak / small / unstable / imprecise)
- The object has a somewhat relevant affordance, but not enough capacity, stability, size, precision, or
strength to work well.

B. Practical invalidity
B1. Destructive or cannibalizing workaround
- The proposal requires damaging, dismantling, cutting, or sacrificing the donor object or another useful
object.

B2. Context / accessibility / process impracticality
- The proposal is too fiddly, inaccessible, complex, unrealistic, or unlikely to be available in the
stated situation.

C. Risk / requirement mismatch
C1. Safety / contamination / damage risk
- The proposal is unsafe, unsanitary, electrically risky, scratch-prone, or likely to damage something.

C2. Violates explicit user constraints or the object's primary purpose
- The proposal ignores explicit constraints or repurposes an object in a way that directly conflicts with
what the user asked to avoid.
```

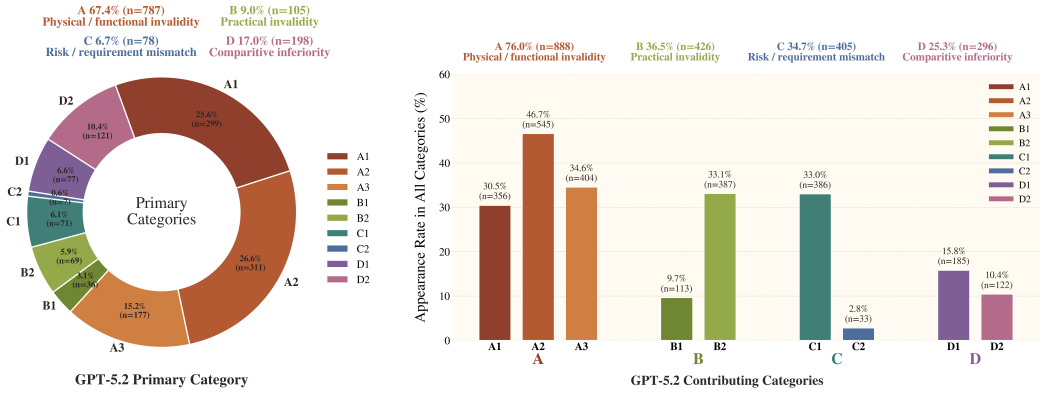


Figure 18: The error attribution analysis of GPT-5.2 respectively on primary category and all contributing reasons.

```
D. Comparative inferiority rather than true failure
D1. Functionally works, but is less preferable in normal use
- The solution is workable, but worse on convenience, accessibility, hygiene, robustness, elegance, or
normal practicality.

D2. Functionally competitive / arguably as good as gold
- The solution is actually reasonable and not clearly worse; the supposed error is mostly preference-
sensitive.

Instructions:
- Base the judgment only on the input sentence.
- Do brief internal reasoning and then return a JSON object only.
- Use only the small category codes like "A1", "B2", "D1".
- "primary_category" must be exactly one code from the taxonomy.
- "all_categories" must be a list of one or more codes from the taxonomy, including the primary category.
- Include multiple codes only when the sentence clearly supports multiple failure reasons.
- Keep "reason" concise, clear, and to the point.
- Do not output markdown, prose outside JSON, or any extra fields.

Return exactly this schema:
{{
  "reason": "explanation",
  "primary_category": "A1",
  "all_categories": ["A1", ...]
}}

Input reason:
{sentence}
```

In the main text, we present the attribution analysis aggregated across all models. In this appendix, we further provide the fine-grained category breakdown for each individual model in Figure 18 to Figure 27. Overall, the patterns remain highly consistent with the main-text results: physical invalidity is the dominant failure reason across models, with similar fine-grained trends appearing in the category distributions.

H.11 Human Study Details

Human evaluation is conducted at multiple stages of benchmark construction and testing, including quality verification of the affordance knowledge base, quality verification of generated benchmark tasks, benchmark-level human evaluation, and validation of the agreement between human and LLM judges. The construction-stage quality checks are described separately in Sections C.2 and D.2, while this subsection focuses on human task-solving behavior and post-hoc evaluation of the annotated gold solutions.

To understand how humans solve our tasks and to assess the quality of the sampled gold solutions, we conduct a human study on 100 tasks. The tasks are balanced across key bench-

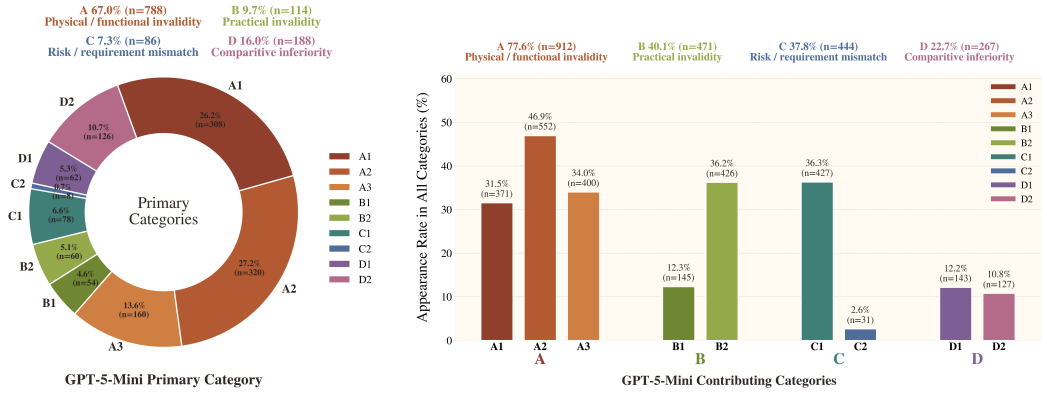


Figure 19: The error attribution analysis of GPT-5-Mini respectively on primary category and all contributing reasons.

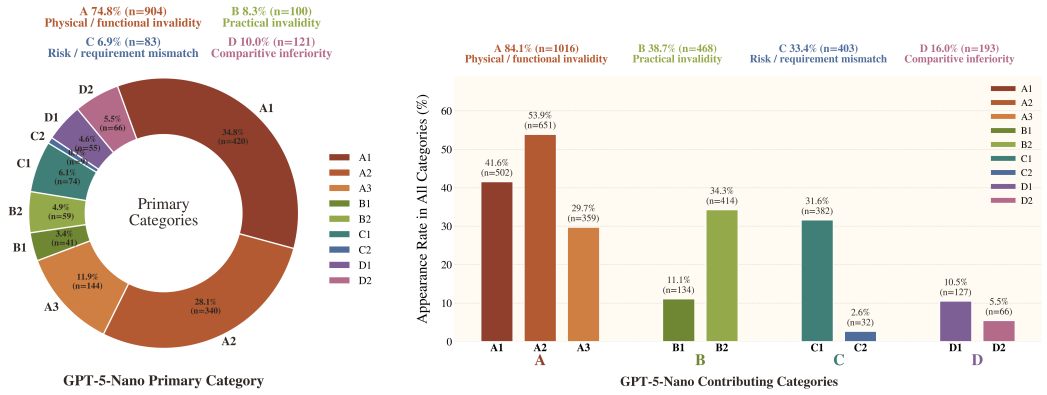


Figure 20: The error attribution analysis of GPT-5-Nano respectively on primary category and all contributing reasons.

mark factors, including entity count, affordance similarity, and affordance typicality level, to provide diverse and representative evaluation cases. We recruit 10 human annotators with computer science backgrounds and adopt a two-stage protocol: **Problem Solving** and **Review**.

In the **problem-solving stage**, annotators are given a task together with a set of candidate entities, each decomposed into parts with associated physical and state attributes. Annotators select the entity and specific part that best accomplish the task based on the provided attributes. This stage mirrors the static evaluation setting used in the main experiments, where the objective is to identify the object part whose physical properties enable the required affordance.

In the **review stage**, annotators compare their selected answer with the annotated gold solution. For each alternative entity–part combination, we provide an attribute-grounded rationale explaining why it is less suitable than the gold answer under the task-specific physical, state, use, environment, and recipient conditions. When an annotator's initial response does not match the gold solution, the annotator reviews this comparative rationale and assesses whether the gold answer is more persuasive. Annotators further rate the gold solution along several dimensions, including physical grounding, feasibility, reasonableness of the use, environment and recipient constraints, and creativity. This review stage provides complementary evidence about whether the sampled gold solutions are physically plausible, well justified, and preferable to the alternatives included in each task.

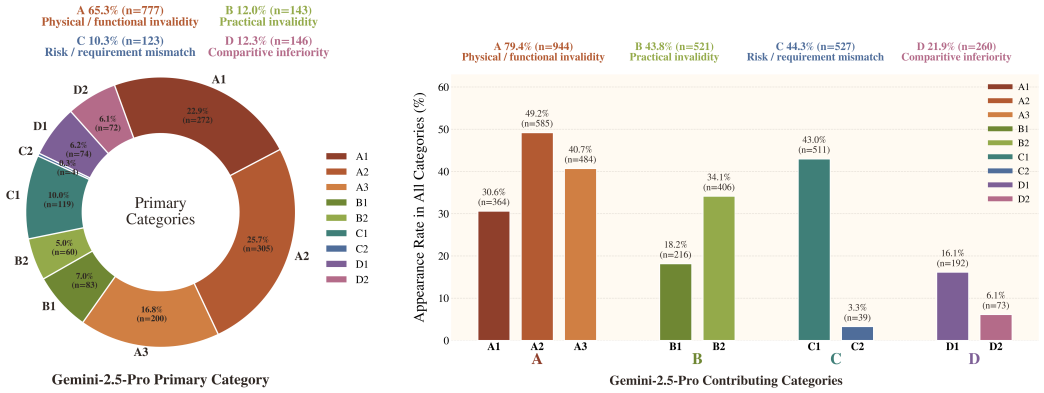


Figure 21: The error attribution analysis of Gemini-2.5-Pro respectively on primary category and all contributing reasons.

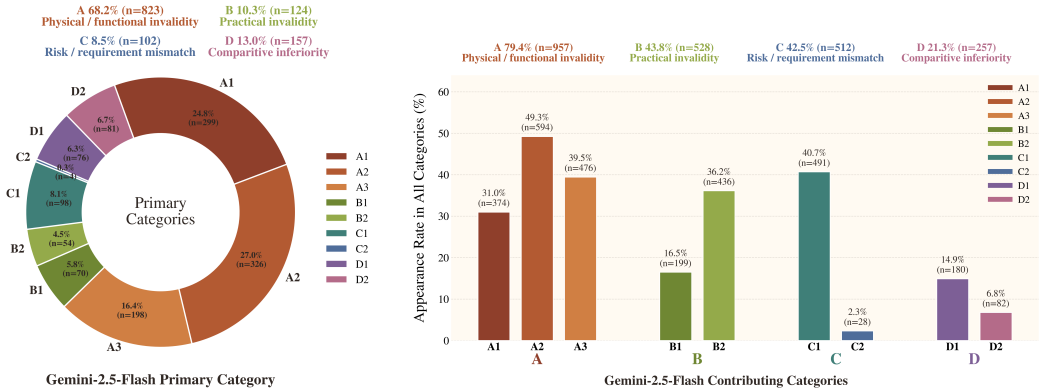


Figure 22: The error attribution analysis of Gemini-2.5-Flash respectively on primary category and all contributing reasons.

Human task-solving performance. Humans achieve a gold correctness of 0.146 and an entity correctness of 0.450, both lower than the best-performing model, Qwen3-32B. We interpret these strict selection results separately from participants' subsequent assessments of the gold solutions. The text-based task context exceeds 8K tokens on average and requires annotators to simultaneously compare multiple entities, their decomposed parts, and detailed physical and state attributes. This makes fine-grained entity–part selection cognitively demanding, particularly because annotators must reason from textual descriptions without direct perceptual access to the objects.

We hypothesize that the text-based presentation imposes substantial cognitive and perceptual load on human participants. The preliminary image-based evaluation described below provides supporting evidence for this interpretation. We nevertheless do not treat cognitive load as the only possible source of human error, since unfamiliar component terminology and residual ambiguity among plausible alternatives may also contribute to task difficulty.

Gold-solution quality and persuasiveness. The review-stage results provide complementary evidence for the quality of the sampled gold solutions. Gold solutions receive average ratings of 4.1533 for physical grounding, 3.8400 for feasibility, 3.8000 for constraint reasonableness, and 3.9200 for creativity. The average gold-persuasiveness score is 3.5800 on a 1–5 Likert scale.

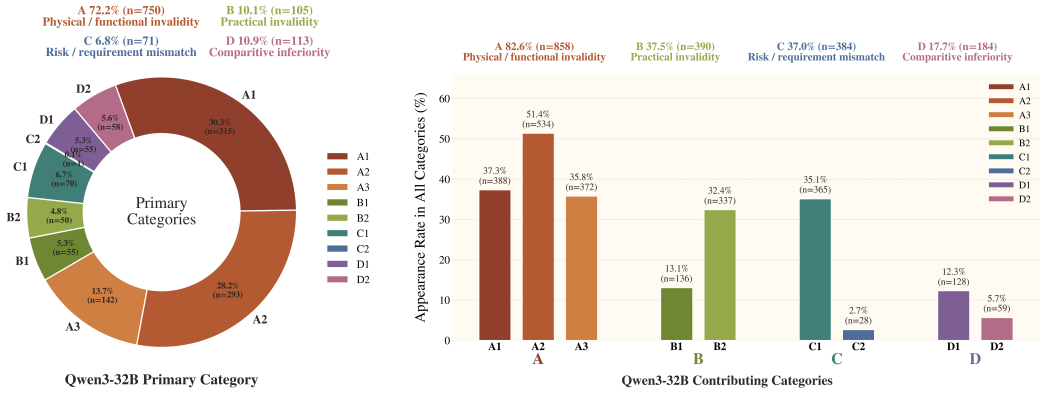


Figure 23: The error attribution analysis of Qwen3-32B respectively on primary category and all contributing reasons.

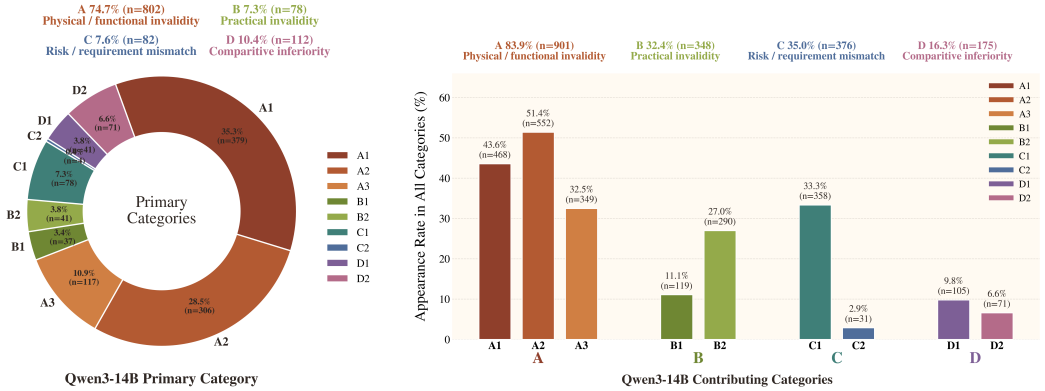


Figure 24: The error attribution analysis of Qwen3-14B respectively on primary category and all contributing reasons.

More than 90% of gold-persuasiveness ratings fall between 3 and 5, indicating generally positive assessments. Thus, even when annotators do not initially identify the gold entity–part combination, they generally acknowledge, after reviewing the comparative rationale, that the annotated gold solution is sufficiently persuasive and well justified. These review-stage judgments provide indirect validation of both the sampled test instances and their annotated gold solutions, while remaining distinct from strict task-solving accuracy.

Inter-annotator agreement. For gold-solution persuasiveness, we compute strict exact-score agreement on the 1–5 Likert scale. Two judgments are counted as agreeing only when annotators assign the identical numerical score. For example, ratings of 1 and 1 or ratings of 5 and 5 are counted as agreement, whereas adjacent ratings such as 4 and 5 are treated as disagreement. This therefore constitutes a fine-grained and deliberately stringent agreement criterion. Under this criterion, inter-annotator agreement is 63.0%.

Differences between adjacent ratings may reflect variation in the perceived degree of persuasiveness rather than disagreement about whether a solution is acceptable. We therefore conduct an additional analysis in which ratings of 1–2 are grouped as *negative* judgments and ratings of 3–5 are grouped as *positive* judgments. Under this binary categorization, inter-annotator agreement increases to 93.0%. This suggests that annotators largely agree on whether a gold solution is acceptable or problematic, even when they differ in the assessed strength of its persuasiveness.

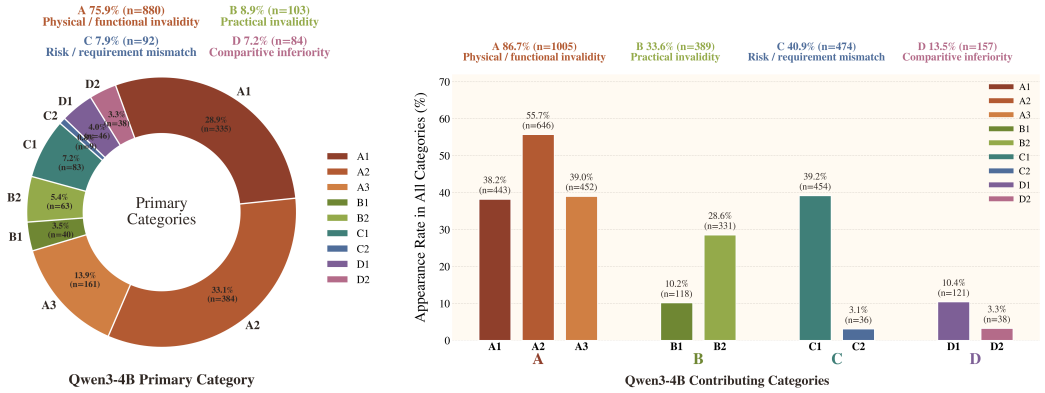


Figure 25: The error attribution analysis of Qwen3-4B respectively on primary category and all contributing reasons.

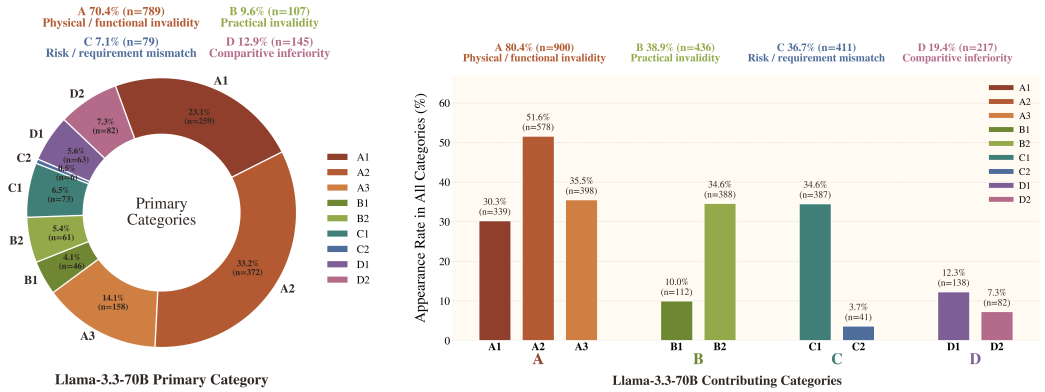


Figure 26: The error attribution analysis of Llama-3.3-70B-Instruct respectively on primary category and all contributing reasons.

Together with the observation that over 90% of all gold-persuasiveness ratings fall in the positive range, these results indicate that annotators generally regard the sampled gold solutions as persuasive and well justified. The lower fine-grained agreement primarily reflects differences in rating severity rather than frequent disagreement about the overall acceptability of the gold solutions.

Preliminary image-based evaluation. To investigate whether the textual presentation contributes to the relatively low human task-solving accuracy, we conduct a preliminary image-based evaluation under a setup largely consistent with the human study in Section 5.5. For a controlled comparison, we use the same sampled questions and the same group of human annotators in the text- and image-based settings. We use Gemini-3-Image-Pro to generate visual representations of the entities, parts, object states, and scenarios. The generation prompts are enriched with the detailed physical and state attributes annotated in our knowledge base, providing fine-grained instructions about the intended object and part configurations. Because the evaluated entities are primarily common everyday objects, they are generally suitable for representation by current image-generation models.

As a quality-control step, we manually inspect a subset of the generated images at both the object and part levels. These inspections indicate that the generated images generally reflect the intended entities, parts, and object states, although fine-grained or non-visual attributes may not always be represented perfectly. The purpose of this experiment is to provide an initial controlled analysis of whether visual references reduce the cognitive burden

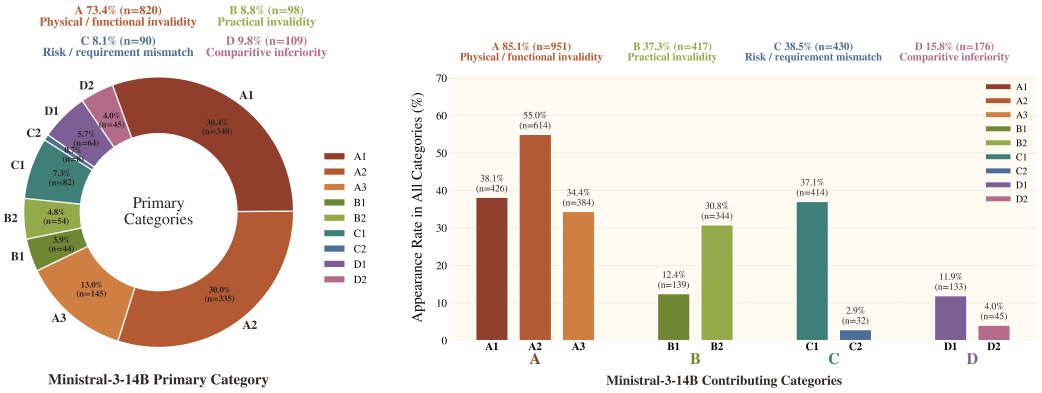


Figure 27: The error attribution analysis of Ministral-3-14B-Reasoning-2512 respectively on primary category and all contributing reasons.

Metric	Human Text Static	Human Image Static	GPT-5.2 Text Static	GPT-5.2 Image Static	GPT-5.2 Image Interactive
Gold Correctness	0.146	0.357	0.210	0.240	0.190
Entity Correctness	0.450	0.677	0.520	0.570	0.440

Table 15: Preliminary comparison of text- and image-based evaluation settings for humans and GPT-5.2. The text- and image-based conditions use the same sampled tasks.

of reasoning over long textual descriptions. Because generated images may introduce representational variation and visual information may be interpreted differently across participants, we treat this experiment as an exploratory study rather than a definitive evaluation.

As shown in Table 15, visual support substantially improves human performance: gold correctness increases from 0.146 to 0.357, while entity correctness increases from 0.450 to 0.677. In comparison, GPT-5.2 improves more modestly in the static setting, from 0.210 to 0.240 in gold correctness and from 0.520 to 0.570 in entity correctness. Its performance decreases under the image-based interactive setting.

The substantially larger improvement for humans is consistent with the hypothesis that visual representations reduce the cognitive and perceptual burden of interpreting and retaining long textual descriptions of entities, parts, and attributes. In contrast, the smaller improvement for GPT-5.2 suggests that the benchmark remains challenging for the model even when some of the textual presentation burden is reduced. Because this study is preliminary and generated images may imperfectly represent fine-grained object states, spatial relations, or non-visual properties, we treat these findings as exploratory evidence rather than a definitive causal demonstration.

Effects of benchmark factors. As shown in Figure 10, human performance is influenced more strongly by the number of distractor entities than by gold affordance frequency or typicality. Human accuracy decreases as the number of distractors increases, since larger candidate sets produce longer and more information-dense contexts and make it harder to track fine-grained part-level attributes. In contrast, model performance is more strongly affected by the frequency and typicality of the target affordance. This difference suggests that human difficulty is driven primarily by the information-processing burden of navigating dense candidate spaces, whereas model difficulty is more closely associated with recognizing rare or less familiar affordances.

Relation to other human quality checks. The present study evaluates human task-solving behavior and post-hoc perceptions of gold-solution quality. It is distinct from the sampled

human audits conducted during KB and task construction. Those audits use separate rubric-based protocols for commonsense validity, groundedness, factual consistency, solution coherence, attribute-grounded rationale, and gold-comparison consistency, and are reported in Sections C.2 and D.2.

I Discussions

Difference of Creativity and Hallucination. Throughout this paper, we define creativity as **grounded** in object attributes and the affordances they induce. In this sense, our notion of creativity is importantly different from creative writing, open-ended design, or innovative research ideation, which may involve freer imagination, sudden insight, or even a degree of productive "hallucination" (Lu et al., 2026). By contrast, the creativity studied here is better understood as **structured creativity**: it still requires novelty, but that novelty must emerge from reasoning over physically plausible object properties and their functional implications. Creative tool use is therefore not about inventing arbitrary solutions, but about connecting the model's goal to the affordances available in the environment in a non-obvious yet grounded way. This distinction is especially important for LLM agents, and even more so for future embodied agents, because tool use is a core capability for them acting in the world. When such agents are grounded in real environments, creativity can directly improve their usefulness to human users by enabling alternative, practical pathways for everyday problem solving under constraints.

Enhancement of Model Creativity. Improving model creativity may require a training objective that differs from the one optimized by much of current reinforcement learning. Existing RL methods, especially unsupervised ones, often emphasize sampling efficiency and empirically lead to distribution sharpening (He et al., 2026), which benefits reliability but can suppress the structured diversity needed for creative problem solving. For instance, although methods like TTRL (Zuo et al., 2025) can improve analytical intelligence in domains like math and coding, their reliance on majority vote as pseudo-labels may further reduce generation diversity and thus work against creativity. Recent work on rewarding unlikeliness (He et al., 2025) points in a promising direction by explicitly encouraging broader exploration, but it has not yet been applied to creative problem-solving settings. Our affordance KB, alternatively, provides one concrete step in this direction: beyond benchmark construction, it can be used to sample thousands of grounded creative problem-solving trajectories, potentially supporting the development of models that learn creative reasoning from data rather than from outcome supervision alone.

J Use of LLMs

In this work, LLMs are used strictly for research support rather than as sources of substantive content. Their use falls into three categories: (i) serving as automatic pipeline annotation helper, (ii) providing tested results on CreativityBench, and (iii) assisting with language refinement during paper writing. For writing support, we used ChatGPT solely to polish text (improving coherence and grammar) while all ideas, logic, results, and technical contributions originate from the authors. To safeguard rigor, we have carefully reviewed all LLM-refined texts to confirm that no hallucinated content was introduced and that the original arguments, findings, and perspectives were faithfully preserved.