From Where Things Are to What They Are For: Benchmarking Spatial–Functional Intelligence in Multimodal LLMs

Supplementary Material

A. Influence of Reasoning Budget

In the last section, we analyzed the open-source model's performance based on its scale without explicitly controlling the reasoning parameter. However, to investigate whether RL-based optimization improves reasoning on SFI-Bench, we now **actively control the *reasoning budget***—the maximum token allowance for the model's reasoning chain—on proprietary models. By varying the budget (see Fig. 8), we can directly assess how changes in reasoning depth affect performance across all SFI-Bench tasks.

Both GPT-5 and Gemini-2.5-Pro show accuracy gains with increased reasoning budgets, indicating that extended reasoning allows for better integration of spatial and functional cues. However, the performance curve flattens beyond approximately 2k tokens, as further exploration reveals that Gemini-2.5-Pro does not utilize reasoning beyond 2k tokens, reaching its limit in reasoning depth at this point.

By examining the reasoning content, we observe that longer reasoning budgets tend to reveal interesting patterns, such as frequent self-checking and the development of complex, ordered plans for task-solving.

Overall, the results suggest that reasoning efficiency, rather than token capacity alone, drives performance. The model's reasoning chain has an upper limit, with a moderate reasoning budget (around 2k tokens) offering an optimal balance between expressivity and stability. Beyond this point, further expansion yields diminishing returns, as good models do not continue reasoning indefinitely but instead reach a point of effective problem-solving.

B. Task Examples

Additional task examples are provided in Fig. 9 and Fig. 10.

C. Detailed Failure Mode Categorization

Below is the detailed failure mode categorization for the analysis introduced in Sec. 4.1:

1. **Visual Perception:** Failures related to object recognition and visual data interpretation. This includes: *Missing objects*, where the model overlooks visible entities; *Object misclassification*, where objects are assigned the wrong labels; *Attribute mislabeling*, where attributes such as color, size, or brand are incorrectly identified; and *Re-identification failure*, where the model mistakenly counts the same object multiple times when viewed from different perspectives. Additionally, *Reflection confusion* occurs when the model misinterprets mirror reflections as real objects.
2. **Spatial Understanding:** Errors related to the model's ability to maintain consistent and accurate spatial representations. This includes: *Positional inconsistency*, where objects shift positions or lose continuity across frames (e.g., teleportation effects); *Geometric misinterpretation*, where the model fails to infer correct geometric relationships between objects (e.g., alignment or linearity); and *Object mislocalization*, where the model places objects in incorrect locations, such as confusing left/right or near/far positioning.
3. **Functional Reasoning:** Failures related to the model's ability to understand functional relationships and perform grounded, compositional reasoning. This includes: *Affordance overgeneralization*, where the model assumes functional relationships based on commonsense (e.g., assuming any remote controls a TV) without verifying the specific context; and *Missing multi-hop reasoning*, where the model fails to complete complex multi-step inferences over functional chain of objects.

D. Data Curation Process Details

This section provides additional details on the dataset construction process, including the metadata format, prompting setup, and human annotation pipeline. Unless otherwise specified, Gemini-2.5-Pro is used as the default MLLM assistant. For clarity, the prompts shown here are lightly streamlined.

Metadata Generation. We begin by generating structured metadata for each video using an MLLM. The metadata includes global scene descriptions, object instances with timestamps, category labels, fine-grained attributes (e.g., color, material, brand), spatial relations, and functional tags. An example (simplified for readability) is shown in Fig. 11. The system prompt used for metadata generation is provided in Fig. 12. Outputs from multiple passes are merged and subsequently verified against the original video.

Task Templates. Given high-quality metadata, task-specific questions are produced using carefully designed templates. The templates for each task are shown in: Fig. 13 (global conditional counting), Fig. 14 (cross-view multi-hop path reasoning), Fig. 15 (layout inference), and Fig. 16

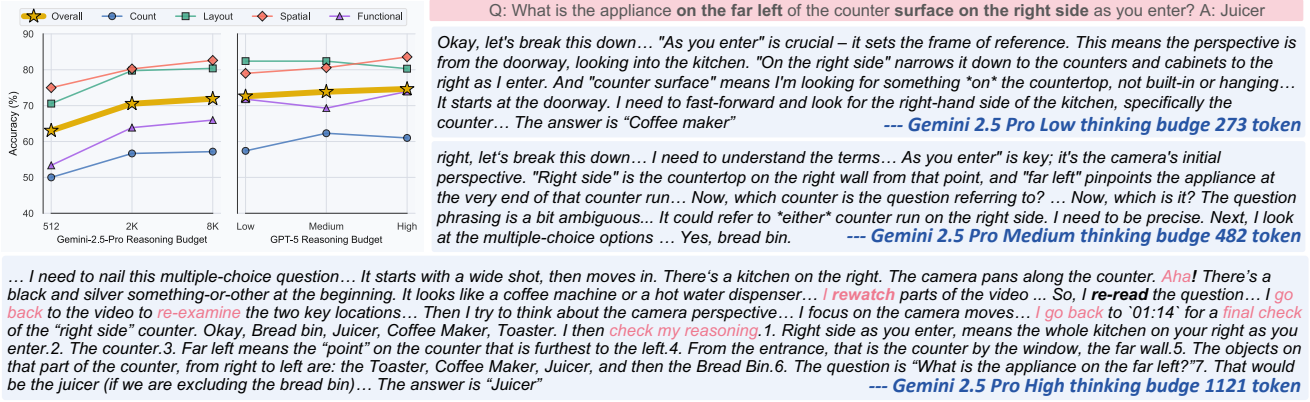

Figure 8. **(Left)** Performance of Gemini-2.5 Pro and GPT-5 under different reasoning budgets. **(Right)** illustrate how increased reasoning budget alters reasoning patterns, higher budgets reveal frequent reasoning behaviors such as self-rechecking and complex reasoning chain.

(functional association). Each template is tailored to ensure consistency, task fidelity, and sufficient reasoning complexity.

Human Verification. The dataset is curated by 11 experienced annotators, all trained machine learning engineers. Each question is manually reviewed by at least one expert to ensure correctness and alignment with the associated video. Annotations are performed using an HTML-based platform that displays the question-generation reasoning chain, relevant objects with timestamps, and the corresponding video interface, as illustrated in Fig. 17.

E. Evaluation Setup

All videos are preprocessed to a fixed resolution of **720p** and **24 fps** before being uploaded to any model. This standardization reduces data size and ensures consistent visual quality across systems, preventing performance differences arising from input variability.

For proprietary models, we evaluate the Qwen3-VL family through the official **Aliyun API**, Gemini models via the **Google Cloud API**, and GPT models using the **OpenAI API**. All API-based experiments were conducted during the first two weeks of **September 2025** to ensure fair comparison under stable model deployments.

For open-source models, we adopt the **VLMEvalKit** framework to provide a unified inference pipeline and standardized evaluation protocol. This setup enables consistent multi-image and video processing across LLaVA-based, InternVL-based, and GLM-based models.

To promote reproducibility, we will release our **evaluation scripts**, **preprocessing pipeline**, and the full **SFI-Bench dataset** upon publication.

Global & Conditional Counting

How many pieces of furniture at or next to the desk contain metal?

Answer: 5

Functional Association

Where is the object that provides the content for the black rectangular device with a red and black stand?
Answer: On right side of the topped desk.

Cross-View Multi-hop Path Reasoning

Starting from the gaming chair, what electronic device is on the tall metal device that is positioned next to the desk?

Answer: headphones

Functional Association

Where is the object with a white frame and burgundy compartment offers a place for both resting and putting things away?

Answer: Opposite the window.

Layout Inference

Which obstacle is blocking the straight-line walking route between the Red Entrance Door and the Hallway Trash Can?

Answer: The Shoe Rack.

Layout Inference

What object should be removed to enable a direct straight path from the Laundry Basket to the Radiator?

Answer: The Standing Fan.

Global & Conditional Counting

How many power outlets are there on the wall opposite the entrance?

Answer: 12

Operational Planning

How do I load paper into the main paper tray for the printer on the desk?

Answer: Pull the main tray (Tray 2) completely out of the printer, open the paper guides, place the paper stack in the tray, adjust the guides to fit, and then reinsert the tray into the printer.

Causal Hypothesis & Trouble Shooting

The printer on the desk is pulling multiple sheets of paper at once from the tray. What's causing this and how do I fix it?

Answer: This is caused by the paper stack or tray condition. To fix it, remove the paper from the tray, flex it, rotate it 180 degrees, and flip it over. Also, ensure the tray is not overfilled and that the paper guides are adjusted correctly.

Global & Conditional Counting

How many heavy-duty staplers are the long-reach type?
Answer: 2

Global & Conditional Counting

How many office supplies with the same function are there on the desk?

Answer: 4

Figure 9. Representative task examples from SFI-Bench. Note that not all videos are equally suitable for every task; for instance, the bottom example depicts a small office with overly simple spatial structure, making it difficult to generate challenging layout or spatial reasoning questions.

Global & Conditional Counting

How many of the wall-mounted reading lights are turned on?

Answer: 2

Cross-View Multi-hop Path Reasoning

What appliance is on the far right of the countertop directly opposite the bed?
Answer: headphones

Answer: Coffee Machine

Layout Inference

Is there a direct path from the purple armchair to the bed?

Answer: Yes

Operational Planning

How do I make a cappuccino using the coffee machine on the counter?

Answer: Connect the suction hose to the spout and place the other end in the milk container. Place a cup under the spout. Turn the rotary control until 'CAPPUCCINO' is displayed, then press the rotary control once to start the process.

Causal Hypothesis & Trouble Shooting

The coffee machine's orange service light above the buttons is on continuously and isn't flashing. What does this mean and what should I do?

Answer: A solid orange light means the machine is clogged with scale and needs to be descaled.

Operational Planning

How do I replace the small, round CMOS battery on the motherboard of the dell desktop computer on the bottom of the right most shelf?

Answer: After opening the case, locate the coin-cell battery on the system board. Press the release latch away from the battery to allow it to pop up, then lift it out. To install the new one, press it into the slot until the latch secures it.

Causal Hypothesis & Trouble Shooting

The power button on the dell desktop computer on the bottom of the right most shelf is blinking amber. What does this mean?

Answer: A blinking amber power light indicates that a problem has occurred with the system board.

Global & Conditional Counting

What is the largest number of desktop computers that are all the same model?

Answer: 12

Operational Planning

How do I open the side cover to access the internal components of the dell desktop computer on the bottom of the right most shelf?

Answer: Lay the computer on its side, then lift the cover-release latch located on the top. Lift the cover upward to a 45-degree angle and remove it from the chassis.

Causal Hypothesis & Trouble Shooting

The diagnostic lights on the front of the dell desktop computer on the bottom of the right most shelf are showing lights 3 and 4 are on. What is the problem and what should I do?

Answer: This light pattern indicates a memory power failure has occurred. To fix it, you should remove all the memory modules, then reinstall one module and restart the computer. If it starts, continue installing modules one by one to identify the faulty one.

Global & Conditional Counting

How many Fujitsu Esprimo desktop computers are present?

Answer: 7

Figure 10. Representative task examples from SFI-Bench. Note that not all videos are equally suitable for every task; for instance, the bottom example depicts a small office with overly simple spatial structure, making it difficult to generate challenging layout or spatial reasoning questions.

```
{
  "video_file": "41069048.mp4",
  "scene_overview": {
    "room": "Bathroom",
    "style": "Modern, clean, functional, hotel-like",
    "palette": "White, grey-blue, chrome"
  },
  "objects": [
    {
      "id": "toilet_001",
      "class": "Toilet",
      "prominent_ts": "00:10",
      "vis_segments": [ {"start": "00:00", "end": "00:02"}, {"start": "00:08", "end": "00:13"} ],
      "attributes": { "type": "Two-piece", "mat": "ceramic", "color": "white" },
      "state": { "lid": "closed", "condition": "clean" },
      "location": "Positioned between trash can and bathtub, against wall",
      "functionality": { "primary": "Waste disposal", "secondary": [] },
      "relations": {
        "type": "functional_group",
        "related": ["toilet_brush_001", "trash_can_001", "toilet_paper_holder_001"]
      }
    },
    {
      "id": "bathtub_001",
      "class": "Bathtub",
      "prominent_ts": "00:19",
      "vis_segments": [ {"start": "00:11", "end": "00:20"} ],
      "attributes": { "type": "Shower-tub combo", "mat": "acrylic", "color": "white" },
      "state": { "fill_level": "empty", "condition": "clean" },
      "location": "Adjacent to sink and toilet, against wall",
      "functionality": { "primary": "Bathing/Showering", "secondary": [] },
      "relations": {
        "type": "integrated_system",
        "core_components": ["shower_system_001", "shower_screen_001", "grab_bar_001"]
      }
    }
    ...
  ],
"spatialLayout": {
      "mainPathway": "The camera moves in a circular path around the small bathroom, starting near the toilet, panning
           up the wall, across the ceiling, down to the shower/tub, across to the sink, and back towards the door.",
      "relativePositions": "The toilet and heated towel rail are on one side of the room. The bathtub and sink are on
          the opposite side. A large mirror is mounted above the sink.",
      "anomaliesOrAbsences": "A DVD case is on the bathroom floor, which is an unusual location for such an item. The
          toilet paper holder is empty."
    },

"functionalEcosystem": {
         "hygiene_and_grooming": {
            "core_objects": [
               "sink_001","toilet_001","bathtub_001","shower_system_001"
            ],
            "supporting_objects": [
            "faucet_001","soap_bar_001","mirror_001","towel_radiator_001","towel_001","towel_002","towel_003","
                toilet_brush_001","trash_can_001"
            ],
            "description": "A complete system for personal hygiene, including washing, bathing, grooming, and waste
                disposal, with all necessary fixtures and accessories present."
         }
      }
}
```


Figure 11. Meta Information Examples.

System Prompt: Meta Information Generation

TASK: Analyze this video and generate a comprehensive, structured JSON representation of the scene and its contents. The goal is to create a rich, machine-readable format suitable for detailed Q&A and object-level analysis. **OUTPUT FORMAT:** `Pure JSON object only` **KEY PRINCIPLES:**

- **Unique Instance Tracking:** Every discrete object (e.g., each individual cup, book, or chair) should be a unique entry in the `objectInventory`. Assign a persistent `instance_id` to each object (e.g., `book_001`, `mug_001`). This ID is crucial for referring to a specific object unambiguously, even if it looks identical to others or reappears after being hidden. For this reason, avoid using a summary "quantity" field.
- **Structured Properties:** Favor structured key-value pairs within an `attributes` object over a single, long description string. This allows for more precise and easily queryable information about each object's visual characteristics.
- **Temporal and State Awareness:** Accurately document when objects are visible and how their state might change. Use `visibility_segments` to track an object's presence over time, which is essential for understanding dynamic scenes with occlusions. For each object, identify the `most_prominent_timestamp` — the single moment when the object appears clearest and largest in the frame.
- **Functionality and Relationship Analysis:** For each object, analyze and describe its primary functionality and its relationships with other objects. Use reasoning to infer functional connections, especially when objects share brands or have complementary purposes.
- **Accuracy and Completeness:** Be as exhaustive and accurate as possible. Capture both large furniture/appliances and smaller, everyday items like books, bottles, cups, toys, remote controls, or electronic gadgets. If any text is clearly legible (e.g., a brand name, a book title), capture it. If a piece of information cannot be determined, use a `null` value for that key.

Here is the required JSON structure. The examples illustrate how to apply the principles above:

```
{
  "sceneOverview": {
    "roomType": "Living Room with Open Kitchen",
    "styleAndAtmosphere": "Modern minimalist, bright with good natural light",
    "mainColorPalette": "Primarily white and natural wood tones, with accents of blue"
  },
  "objectInventory": [
    {
      "instance_id": "phone_charger_001",
      "object_class": "Phone Charger",
      "visibility_segments": [ { "start": "00:08", "end": "01:25" } ],
      "attributes": {
        "type": "USB-C cable with wall adapter",
        "color": "white",
        "cable_length": "1 meter",
        "brandAndModel": "Apple 20W USB-C Power Adapter",
        "recognizedText": "Apple"
      },
      "state": {
        "connection_status": "plugged into wall outlet",
        "cable_condition": "coiled neatly"
      },
      "relational_location": "On the nightstand next to the bed, near the wall outlet.",
      "functionality_relation": {
        "target_objects": ["iphone_001", "ipad_001"],
        "reasoning": "Apple charger is specifically designed for Apple devices, and there's an iPhone visible on
            the nightstand with the same charging port",
        "relationship_type": "power_supply",
        "compatibility": "brand_specific"
      }
    }
  ],
  "spatialLayout": {
    "mainPathway": "The path from the room's entrance to the sofa area is clear, but a floor lamp partially
        obstructs the path between the sofa and the balcony.",
    "relativePositions": "The bookshelf is to the left of the sofa. The window is on the south wall of the room
        .",
    "anomaliesOrAbsences": "A dumbbell is visible on the kitchen counter, which is unusual for a kitchen."
  },
  "functionalEcosystem": {
    "entertainment_zone": {
      "core_objects": ["tv_001", "sofa_001", "remote_control_001"],
      "description": "Integrated entertainment setup where TV, seating, and control devices work together"
    },
    "connectivity_infrastructure": {
      "core_objects": ["router_001"],
      "dependent_objects": ["smart_tv_001", "laptop_001", "smartphone_001"],
      "description": "Network infrastructure enabling smart device functionality throughout the space"
    }
  },
}
```


Figure 12. The system prompt used to drive the MLLM for meta information generation.

Task Template 1: Global Conditional Counting

ROLE: You are an AI assistant specializing in VQA dataset creation. Your task is to generate a diverse set of natural, high-quality question–answer pairs from structured JSON annotations. **OBJECTIVE:** Create questions clearly answerable via exact lookups. **INPUT:** Use the provided {{`OBJECT_INVENTORY`}} as the single source of truth. **2. CRITICAL**

RULES (Selected):
1. **Exact Attribute Matching:** Questions must be based on full, exact values. No substrings.
2. **Answer Value $\geq$ 2:** The integer answer must be 2 or more.
3. **Object-Class Specificity:** MUST specify a clear class (e.g., "chairs", "bottles"). Avoid generic "objects".
4. **Meaningful Filtering:** Conditions must strictly reduce the count (Answer $<$ Total objects of that class).
5. **Scene-Specific Focus:** Questions should be grounded in the specific scene, not universal.
6. **Natural Phrasing:** Do not expose JSON structure (e.g., use "wooden" instead of "material: wood").

3. DIFFICULTY LEVEL DEFINITIONS:
AVOID THESE : Overly broad ("How many items are good?"); Color-only ("What is red?"); Technical jargon ("Made of MDF?"); Ambiguous categories ("types of furniture").
PREFER THESE : Class-specific ("wooden chairs"); Contextual ("lamps turned on"); Brand-specific ("Apple laptops"); Material focused ("glass bottles").

Level 1: Single-Condition Counting *Logic: Count instances of an object_class matching ONE specific attribute.*
- **Q:** "How many of the wine bottles are still sealed?"
- **Rationale:** Filters `object_class: "wine_bottle"` where `attributes.state: "sealed"`.
- **Q:** "How many wooden bar tools are there?"
- **Rationale:** Filters `object_class: "bar_tool"` where `attributes.material: "wood"`.

Level 2: Multi-Condition Counting *Logic: Combine multiple constraints (AND / OR) for sequential filtering.*
- **Q:** "How many pieces of glassware are both clear and clean?" (AND)
- **Rationale:** Filters `glassware` where `color: "clear"` AND `state: "clean"`.
- **Q:** "How many bottles are either 'mostly_full' or 'half_full'?" (OR)
- **Rationale:** Filters `bottle` where `state` IN `["mostly_full", "half_full"]`.

Level 3: Complex Aggregation & Analysis *Logic: Grouping, comparison, or set operations beyond simple filtering.*
- **Q:** "What is the largest number of wine bottles that come from the same brand?"
- **Rationale:** Groups `wine_bottle` by `brandAndModel`, returns size of the largest group.
- **Q:** "How many more sealed wine bottles are there than unsealed ones?"
- **Rationale:** Computes (Count A [sealed]) - (Count B [not sealed]).

4. OUTPUT FORMAT:

```
[
  {
    "question_id": "scene_id_XX_Y_cnt",
    "level": <1, 2, or 3>,
    "question_text": "<Natural question>",
    "question_type": "count",
    "rationale": "<Step-by-step explanation of filtering logic>",
    "visibility_segments": [ ... ],
    "generated_by": "llm-creative"
  }
]
```

Now, generate a list of diverse and interesting Questions based on the objects and corresponding attributes that strictly follow the rules.

Figure 13. The system prompt for generating Global Conditional Counting tasks.

Task Template 2: Cross-View Multi-hop Path Reasoning

ROLE: You are an AI expert in spatial reasoning. Your task is to generate complex, path-dependent Question-Answer pairs based on video JSON annotations. **TARGET TASK:** *Task 1.2: Cross-View & Path-Dependent Localization*. Questions must test a model's ability to construct a 3D mental map and follow multi-step spatial chains without explicit target naming.

1. CORE INSTRUCTIONS (The "Path" Logic):

- **Select a Target:** The answer object (hidden from the question text).
- **Identify Landmarks:** Select anchor objects to start the reasoning chain.
- **Construct the Chain:** Create a logical path (e.g., Landmark → Spatial Relation → Intermediate Object → Spatial Relation → Target).
- **Implicit Inference:** Use scene context (e.g., "opposite the sink") rather than just explicit 'relational$_{l}ocation`fields$.
- **MANDATORY:** Include `visibility_segments` for ALL referenced objects (landmarks + target).

2. CRITICAL CONSTRAINTS & ANTI-PATTERNS:

- **Minimal Descriptions:** Use generic terms ("the machine", "the container") instead of specific attributes ("the Samsung washer") to force spatial reasoning.
- **Avoid Direct Targeting ():** Do NOT describe unique attributes that identify the target without spatial logic.
- **Avoid Breaking the Chain ():** Do NOT explicitly state a landmark's absolute location (e.g., "On the floor, there is a hamper..."). The location must be found relative to other objects.

3. FEW-SHOT EXAMPLES (Study the Rationale):

Example 1: Easy (2 Hops)

```
{
  "question_text": "What object is mounted on the wall above the appliance that sits to the right of the
      entrance?",
  "answer": "the heated towel rail",
  "rationale": [
    "1. Identify anchor: the entrance (door_001).",
    "2. Locate appliance to its right: washing_machine_001.",
    "3. Find object mounted above it: heated_towel_rail_001."
  ]
}
```

Example 2: Medium (3 Hops - Nested Containers)

```
{
  "question_text": "What item is sitting on the edge of the fixture that is located next to the wall-mounted
      toilet?",
  "answer": "the bath toy",
  "rationale": [
    "1. Identify anchor: toilet_001.",
    "2. Locate fixture next to it: bathtub_001.",
    "3. Find target on the edge of bathtub: bath_toy_002."
  ]
}
```

Example 3: Hard (Inferred Layout & Virtual Path)

```
{
  "question_text": "Begin at the framed poster in the hallway. If you pass through the nearby door, what piece
      of furniture has a covered container partially underneath it?",
  "answer": "the sink vanity",
  "rationale": [
    "1. Identify anchor: movie_poster_002 (Hallway).",
    "2. Virtual path: Pass through nearby door_001.",
    "3. Locate intermediate: cat_litter_box_001 (covered container).",
    "4. Identify relation: It is under the sink_vanity_001."
  ]
}
```

Figure 14. The system prompt for generating Cross-View Multi-hop Path Reasoning tasks.

Task Template 3: Layout Inference

ROLE: Generate spatial layout questions based on the following triplet relationships from a video annotation. Layout Triplets:LAYOUT_TRIPLETS Object Inventory (for visibility segments): OBJECT_INVENTORY Each triplet describes a spatial relationship between three objects (object1, object2, object3) where object2 acts as an obstacle or barrier between object1 and object3.

1. INPUT DATA LOGIC:

- **Layout Triplets:** You will receive triplets in the format $(Object_1, Object_2, Object_3)$, where $Object_2$ **acts as an obstacle/barrier** between $Object_1$ and $Object_3$.
- **Object Inventory:** Use this to extract temporal 'visibility$_segments$'.

2. QUESTION CATEGORIES (Vary Difficulty 1–3): Generate questions covering these specific spatial reasoning types:

1. **Direct Path:** Test immediate accessibility (e.g., "Can you walk directly from the sink to the toilet?").
2. **Obstacle Identification:** Identify the blocker (e.g., "What object blocks the direct path from the trash can to the bathtub?").
3. **Alternative Path:** Path planning (e.g., "If you want to go from the DVD case to the trash can, what do you need to go around?").
4. **Multi-step Navigation:** Complex routing (e.g., "To reach the door from the sink, which objects would you need to navigate around?").
5. **Spatial Positioning:** Relative location logic (e.g., "Which object is positioned between the toilet and the towel rack?").

3. MANDATORY CONSTRAINTS:

- **Data Grounding:** Answers must be logically derived strictly from the provided triplet relationships.
- **Visibility Data:** For EVERY question, you **MUST** include the `visibility_segments` for all objects referenced. This is critical for temporal validation.
- **Naming:** Use object names exactly as they appear in the triplets.

4. REQUIRED JSON OUTPUT FORMAT:

```
[
  {
    "question_id": "[auto-generated]",
    "question_text": "What object blocks the direct path from the trash can to the bathtub?",
    "answer": "The Toilet",
    "rationale": "Based on triplet (Trash Can, Toilet, Bathtub), the Toilet is the obstacle.",
    "question_type": "layout-reasoning",
    "sub_question_type": "obstacle_identification",
    "level": 2,
    "visibility_segments": [
      {
        "object_id": "trash_can_001",
        "visibility_segments": [[10.5, 15.2], [20.1, 25.0]]
      },
      {
        "object_id": "toilet_001",
        "visibility_segments": [[0.0, 30.0]]
      },
      {
        "object_id": "bathtub_001",
        "visibility_segments": [[5.0, 20.0]]
      }
    ],
    "generated_by": "llm-creative"
  },
  {
    "question_id": "[auto-generated]",
    "question_text": "Can you walk directly from the sink to the door?",
    "answer": "No, the Washing Machine blocks the path.",
    "rationale": "Triplet (Sink, Washing Machine, Door) indicates an obstruction.",
    "question_type": "layout-reasoning",
    "sub_question_type": "direct_path",
    "level": 1,
    "visibility_segments": [ ... ]
  }
]
```

Figure 15. The system prompt for generating Layout Inference tasks.

Task Template 4: Functional Association

ROLE: You are an AI assistant specializing in VQA dataset creation. Your task is to generate natural, high-quality questions focusing on **multi-object functional relationships**.

OBJECTIVE: Create questions that require understanding active functional interactions (data, power, control) between at least 2 objects. Questions must be impossible to answer without video understanding.

1. CRITICAL RULES (Strict Constraints):

1. **Multi-Object Functional Only:** Focus EXCLUSIVELY on active relationships (data processing, signal transmission, power supply). **NEVER** use trivial spatial relations like "support", "rests_on", "beside", or "near".
2. **Physical Descriptions Only:** Reference objects ONLY by neutral attributes (color, shape, material). **NEVER** mention function or purpose (avoid "display", "storage", "cooking").
3. **Minimal Target Identification:** When asking about the target, use generic terms ("What device..."). **NEVER** describe the target's visual features in the question (e.g., do NOT say "What large white object...").
4. **Video-Dependent:** Questions must require understanding *interaction*, not just appearance.

2. QUESTION TYPES & EXAMPLES:

Level 1: Object Processing/Control (Input/Output/Storage)

- *Logic: Ask which object processes input from or controls another.*
- **Example (Input):** Q: "What receives input from that grey and black device on the desk?"
  A: "The black computer tower on the floor."
  Rationale: Identifies mouse (grey/black) → functionality_relation → PC tower.
- **Example (Output):** Q: "What device sends signals to those two black rectangular objects?"
  A: "The black computer tower on the floor."
  Rationale: Identifies monitors → functionality_relation → PC tower.

Level 2: Spatial-Functional Context

- *Logic: Ask about objects based on their functional ecosystem.*
- **Example (Ecosystem):** Q: "What two objects are positioned together in this room?"
  A: "The green recycling bin and black trash can."
  Rationale: Spatial proximity + distinctive colors → functional pair.
- **Example (Integration):** Q: "What device connects to those black rectangular objects and that black keyboard?"
  A: "The black computer case on the floor."

3. REQUIRED OUTPUT FORMAT:

```
[
  {
    "question_id": "scene_id_XX",
    "level": <1, 2, or 3>,
    "question_type": "Multi_Object_Relationship",
    "question_text": "What object works with that blue fabric item at the desk?",
    "answer": "The light wood cabinet with silver handles under the desk",
    "rationale": "Identified 'blue fabric item' as office_chair_001. Used functionality_relation to find
        associated filing_cabinet_001.",
    "objects_involved": [
      {
        "instance_id": "office_chair_001",
        "attributes": "black frame, blue fabric upholstery",
        "most_prominent_timestamp": "00:01"
      },
      {
        "instance_id": "filing_cabinet_001",
        "attributes": "light wood color, silver handles",
        "most_prominent_timestamp": "00:02"
      }
    ],
    "generated_by": "llm-creative"
  }
]
```

4. KEY REQUIREMENTS SUMMARY:

- Use `functionality_relation`, `relational_location`, and `functionalEcosystem` data extensively.
- Focus on: control, processing, data transmission, signal flow, power supply.
- Include an `objects_involved` array listing all relevant objects.

Figure 16. The system prompt for generating Functional Association tasks.

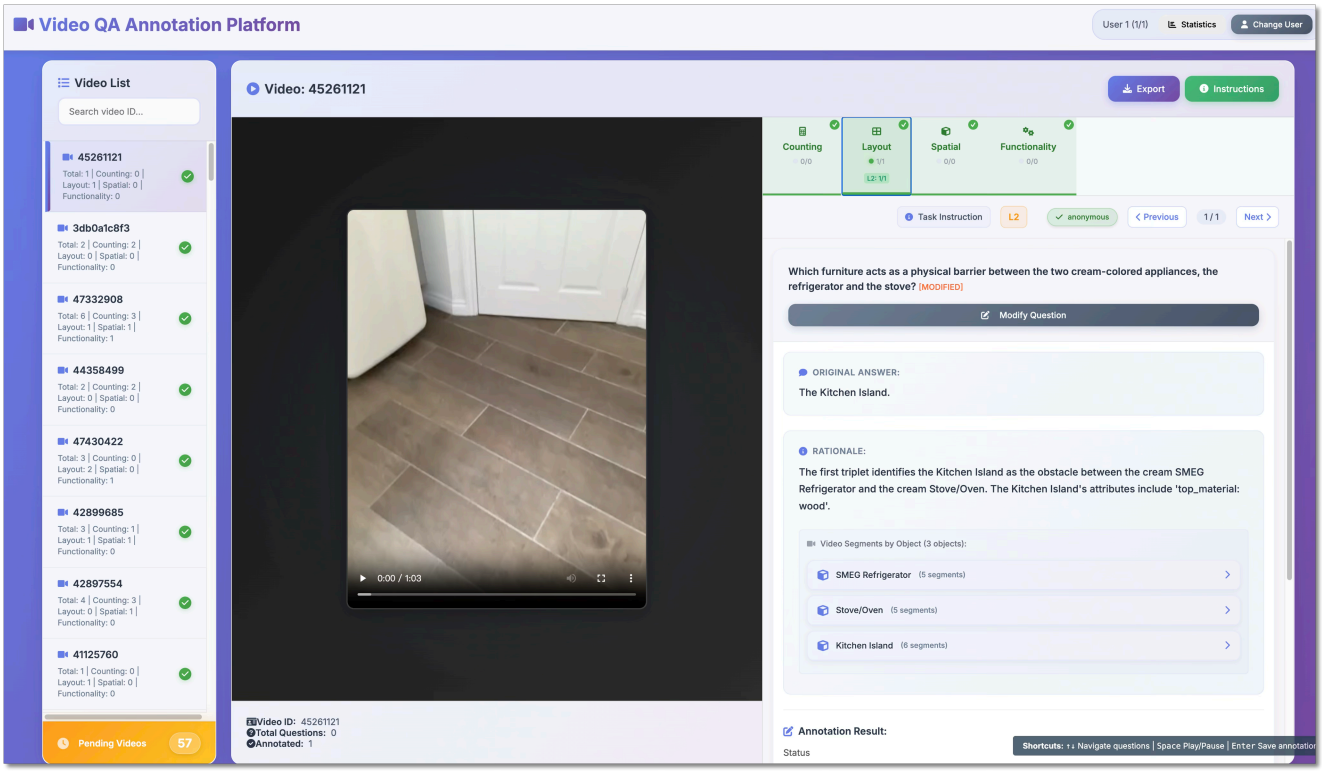

Figure 17. Annotation Platform for human annotaion questions and answers.