NLP COURSE PROJECT 2025

Neuro-Symbolic Commonsense Reasoner

Advanced Task: Visual Reasoning via VLM Scene Graphs

BITS Pilani, K. K. Birla Goa Campus

Agenda

Problem Statement: Bridging Vision & Logic

Related Work: The ESCA Framework

Methodology: The "Eye-Brain-Judge" Pipeline

Architecture: DETR + LLaVA Integration

Conclusion

Future Directions

The Challenge: MLLM Hallucination & Perception Failures

The Core Problem with Pure MLLMs

Existing MLLMs do not reliably capture fine-grained links between low-level features and high-level textual semantics, leading to weak grounding and inaccurate perception. (ESCA)

Object Hallucination

VLMs invent objects or attributes that are statistically likely but not present in the image, or misidentify them.

Geometric Ungrounding

Spatial relationships are based on learned correlation (e.g., "cup on table") rather than hard geometric evidence.

Flawed Symbolic Input

These errors result in a noisy, unreliable set of predicates, causing the symbolic reasoning chain to fail.

Our Solution

We require a Neuro-Symbolic Pipeline using geometric checks (The Judge) to enforce physical grounding and ensure reliability.

REFERENCE WORK

State-of-the-Art Context: ESCA

Our project is inspired by the ESCA (Embodied Scene-Graph Contextualized Agents) framework, which addresses fine-grained perception.

Concept Extraction

Identifying relevant entities from instructions (e.g., "toaster", "bread") using a large MLLM like InternVL (open source) or Gemini/ChatGPT.

Entity Identification

Using GroundingDINO + SAM2 for pixel-perfect object masks.

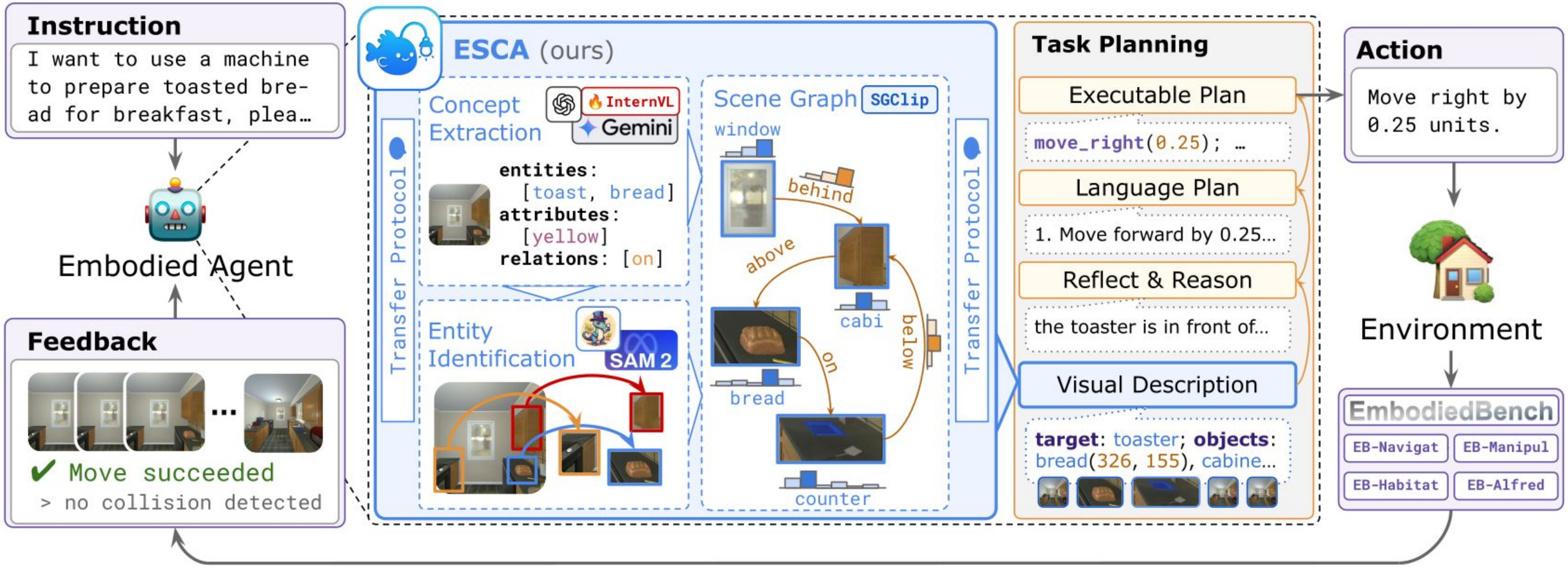
Scene Graph Generator (SGClip)

A fine-tuned CLIP model predicting attributes and relations.

GOLD STANDARD

Reference: The ESCA Pipeline

The state-of-the-art approach combines three steps to achieve "Pixel-Perfect" reasoning. This is the heavy-compute architecture we aim to approximate.



Source: Huang et al., “ESCA: Contextualizing Embodied Agents via Scene-Graph Generation,” arXiv:2510.15963, 2025.

Compute Constraints & Adaptation

Replicating state-of-the-art neuro-symbolic pipelines requires massive compute. Our environment imposed strict boundaries, shaping our architectural choices.

Resource Limit

Memory Bottleneck:
Max **30 GB VRAM** available. Loading multiple large models (e.g., InternVL + GroundingDINO) simultaneously triggers immediate OOM.

Finetuning CLIP is not Feasible:
Replicating SGClip by finetuning CLIP using the existing LASER pipeline demands substantial computational resources.

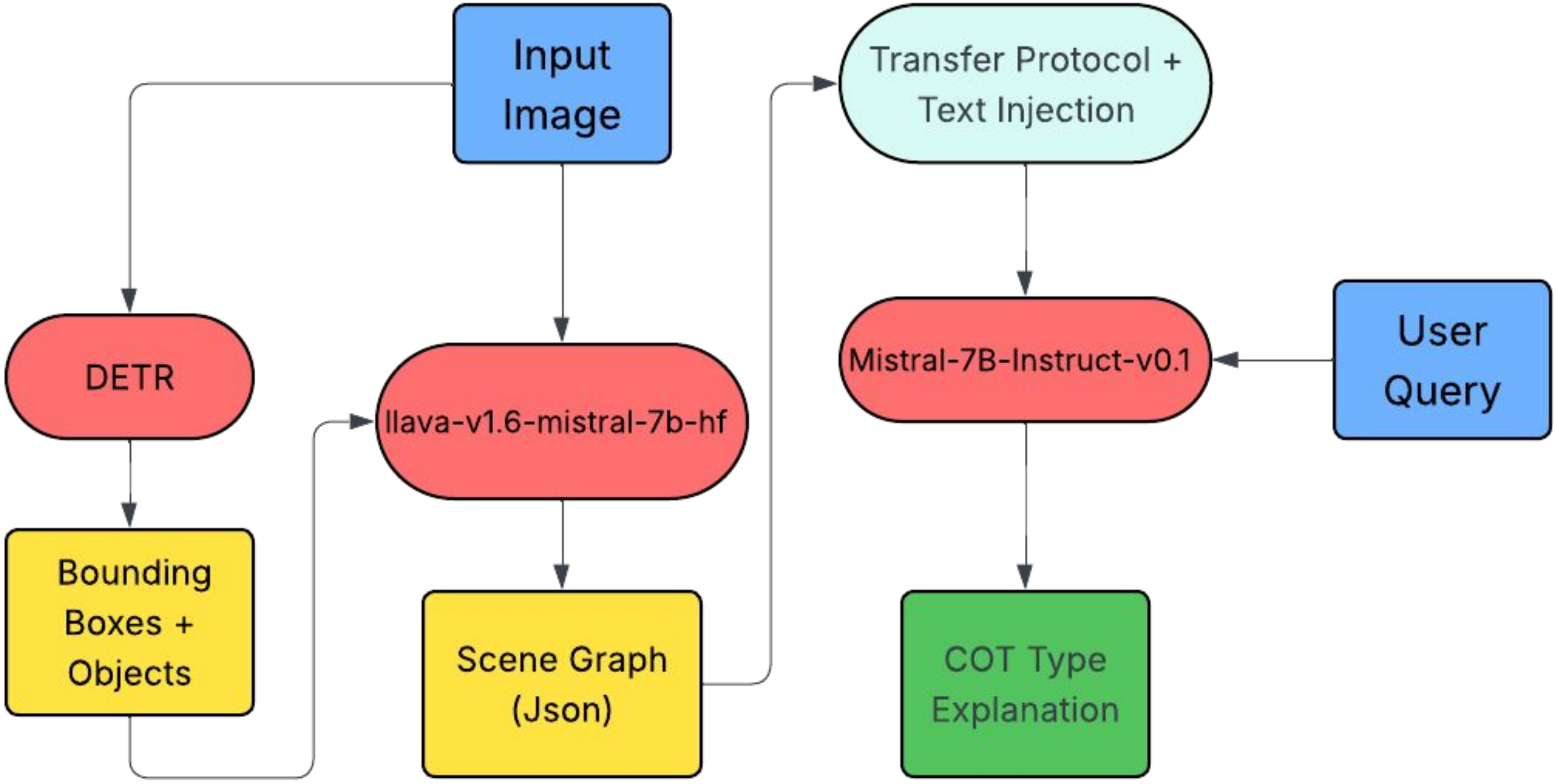
Small Models:
Using smaller Multimodal Large Language Models (MLLMs) was necessary due to memory constraints. These models are known to be poor at reasoning, prone to hallucination, and have limited context windows.

Architectural Pivot

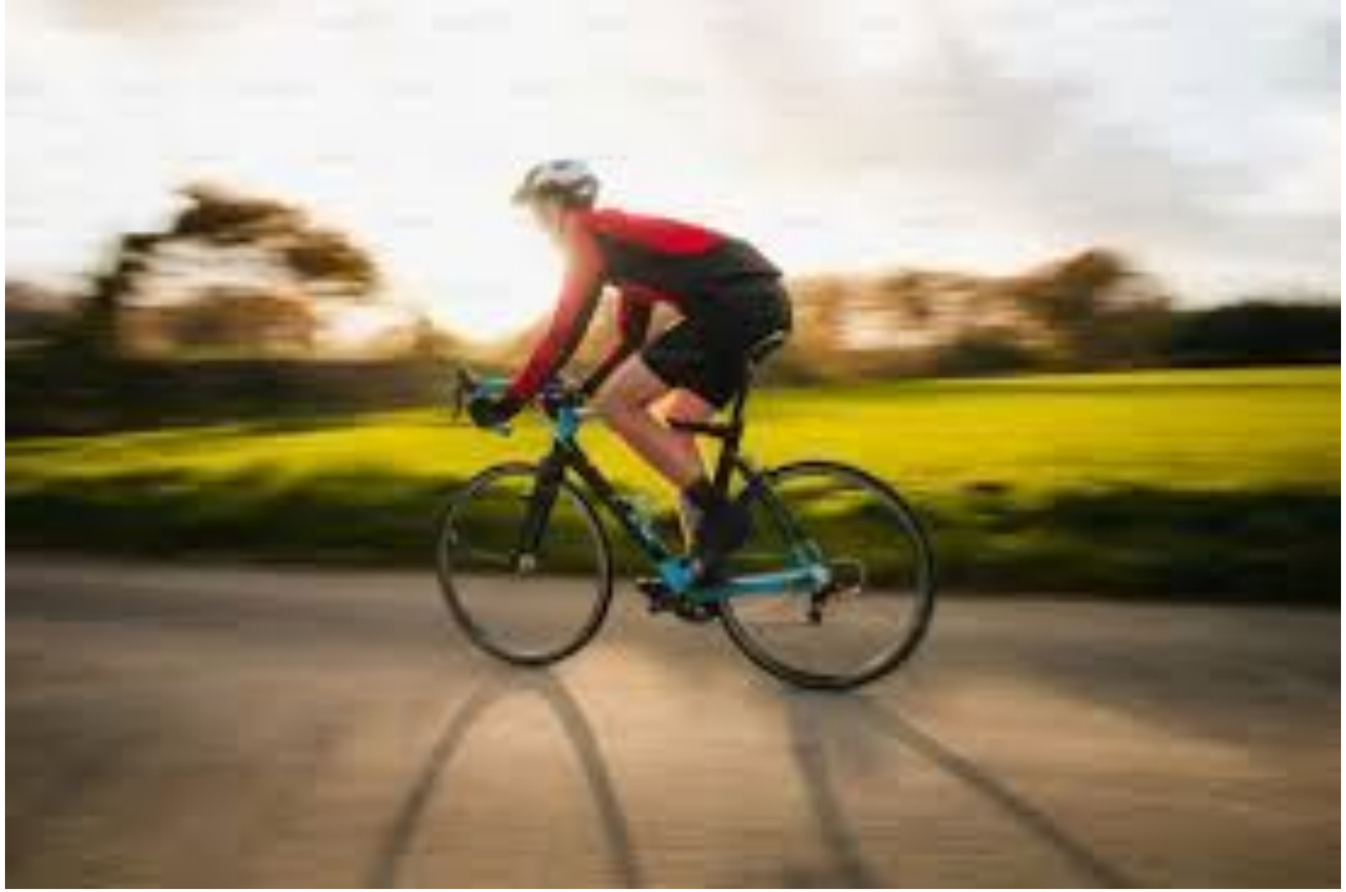
Model Cap: 7B Parameters
Restricted model selection to efficient 7-8B variants (Mistral, LLaVA) rather than heavier 34B+ models.

Sequential Pipeline:
Serial Execution required. We load one model, perform inference, offload weights, and clear cache before loading the next.

Our Simplified ESCA-Like Pipeline



Pipeline Evolution : Output Quality



OLD PIPELINE (Naive):

```
                          // DINO Detected: bicycle, rider,
road, sky, helmet ... { "objects": { "obj1": {"name":
"bicycle", "attributes": ["blue", "metal", "moving"]} },
"relationships": [ {"subject": "obj1", "predicate": "in
front of", "object": "obj2", "attributes": ["green",
"grassy"]} ] } // ⚠ Result: Missing subjects,
hallucinated "obj2", sparse graph.
```

NEW PIPELINE (Eye-Brain-Judge):

```
                                  { "objects": { "person":
{ "name": "Person", "attributes": ["color: red and
black", "size: medium", "state: riding"] }, "bicycle": {
"name": "Bicycle", "attributes": ["color: blue",
"material: metal", "state: moving"] } }, "relationships":
[ { "subject": "person", "predicate": "on", "object":
"bicycle" } ] } // ✅ Result: Grounded entities, correct
attribution, valid relation.
```

The "Eye-Brain-Judge" Pipeline

We decouple the reasoning process into three distinct, specialized phases to maximize accuracy and minimize hallucinations.

Phase 1: The Eye

Localization

DETR (ResNet-50)
Hard Coordinates

Phase 2: The Brain

Semantics

LLaVA v1.6
Context-Aware Prompting

Phase 3: The Judge

Verification

Mistral-7b uses the scene graph to decide the answer to the question

Phase 1: The Eye (DETR)

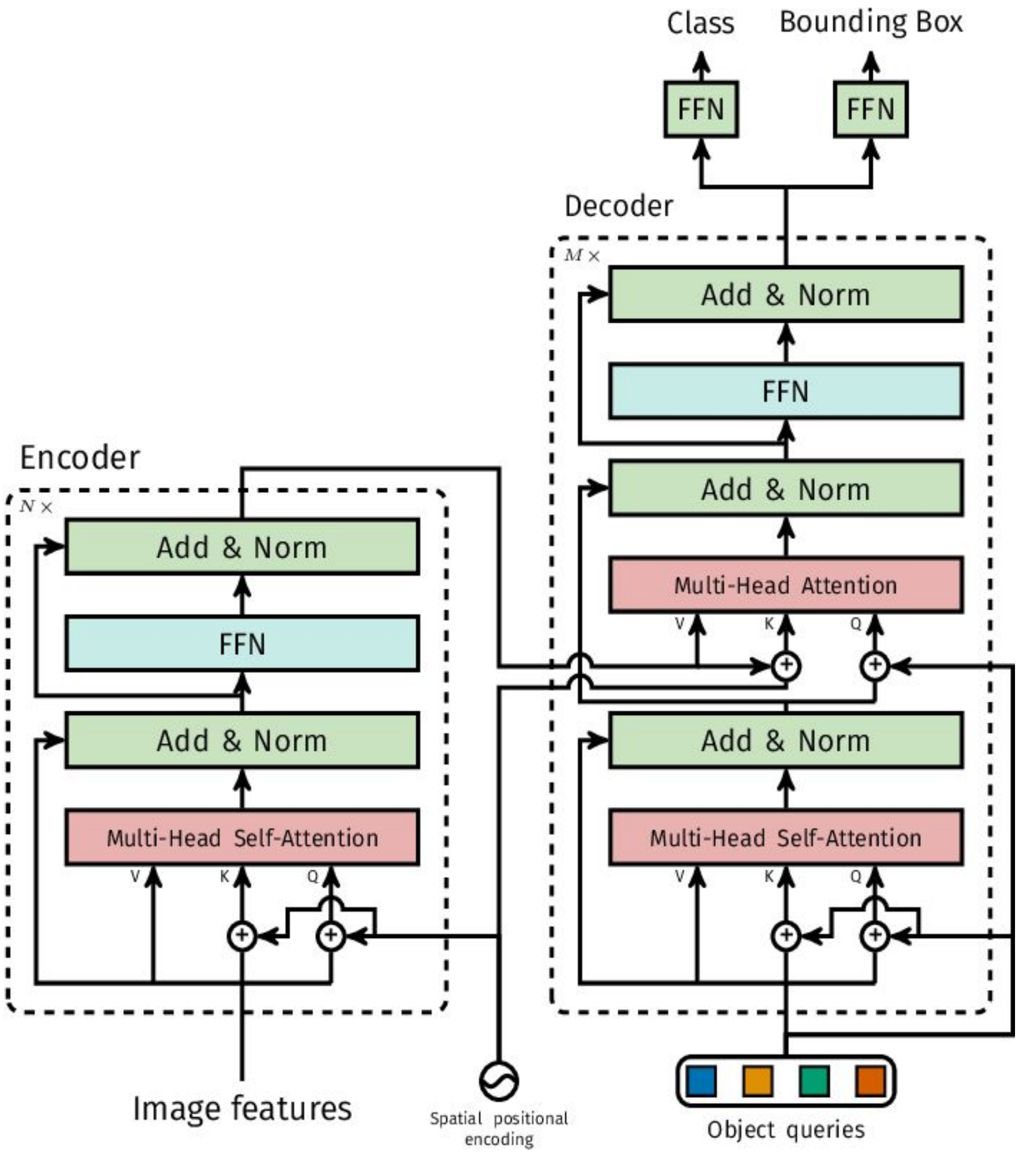
Detection Transformer

We use DETR (Facebook's DEtection TRansformer) for object localization.

Unlike VLMs, DETR provides "hard" mathematical bounding boxes.

- **Role:** Establish Ground Truth.
- **Why DETR over YOLO?**
- The advantage of DETR (and its modern variants) over traditional YOLO models is its superior utilization of **global context awareness** via the Transformer architecture.
- This allows DETR to process the **entire image feature map simultaneously** using multi-head self-attention, which models **long-range dependencies** between objects.

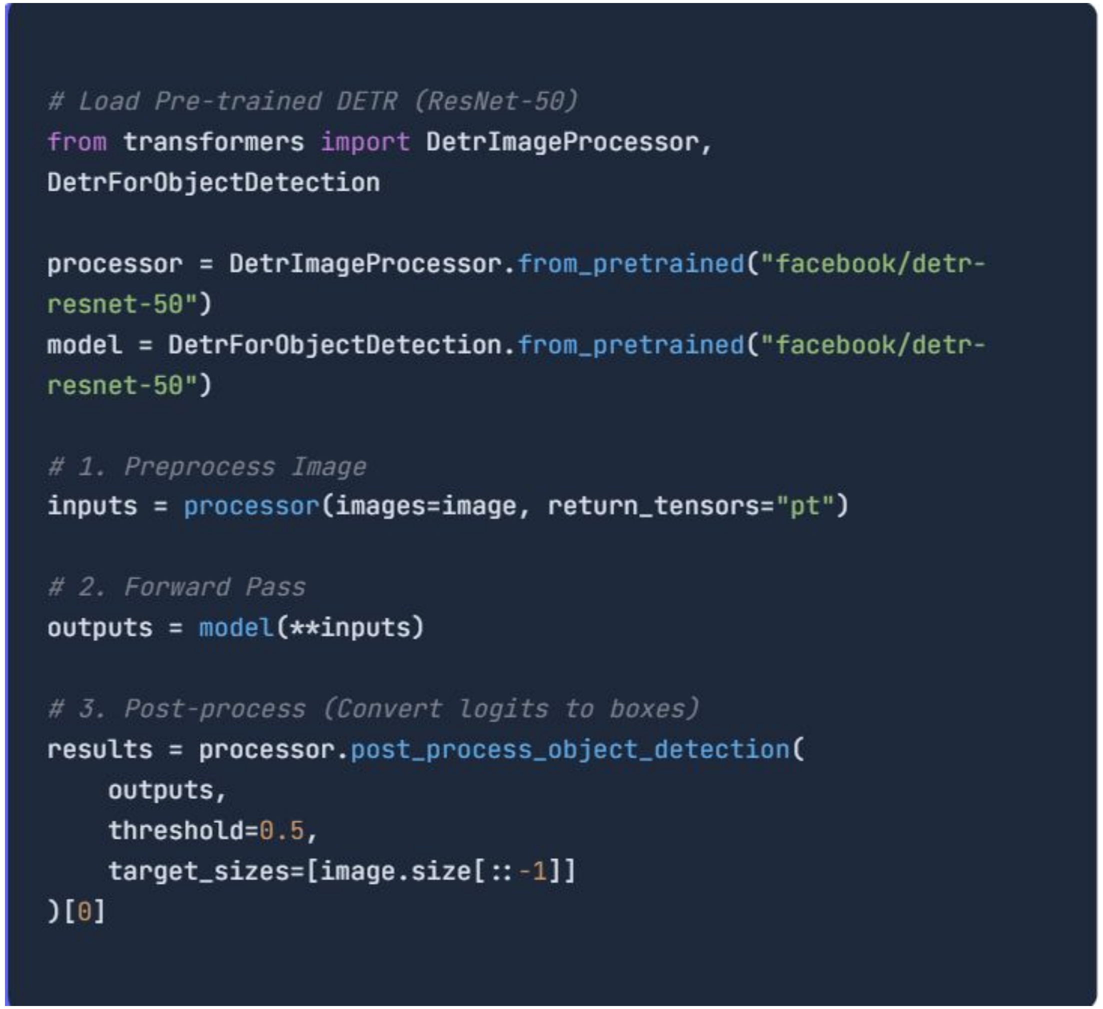
Output: Precise Bounding Boxes [x, y, w, h].



Object Detection

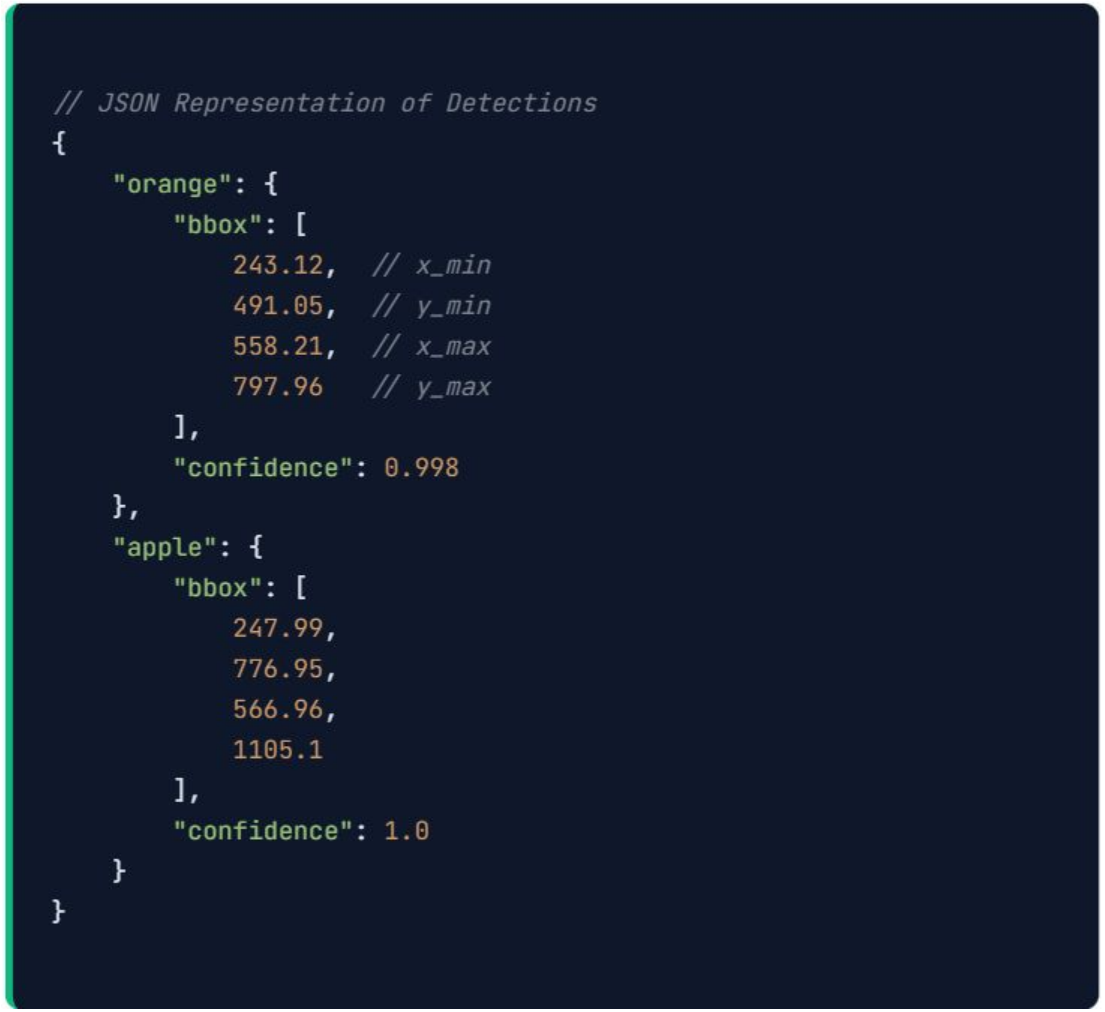
Leveraging HuggingFace Transformers to extract precise geometric ground truth.

1. Model Inference



```python
# Load Pre-trained DETR (ResNet-50)
from transformers import DetrImageProcessor,
DetrForObjectDetection

processor = DetrImageProcessor.from_pretrained("facebook/detr-
resnet-50")
model = DetrForObjectDetection.from_pretrained("facebook/detr-
resnet-50")

# 1. Preprocess Image
inputs = processor(images=image, return_tensors="pt")

# 2. Forward Pass
outputs = model(**inputs)

# 3. Post-process (Convert logits to boxes)
results = processor.post_process_object_detection(
    outputs,
    threshold=0.5,
    target_sizes=[image.size[::-1]]
)[0]
```

2. Structured Output



```
// JSON Representation of Detections
{
    "orange": {
        "bbox": [
            243.12,  // x_min
            491.05,  // y_min
            558.21,  // x_max
            797.96   // y_max
        ],
        "confidence": 0.998
    },
    "apple": {
        "bbox": [
            247.99,
            776.95,
            566.96,
            1105.1
        ],
        "confidence": 1.0
    }
}
```

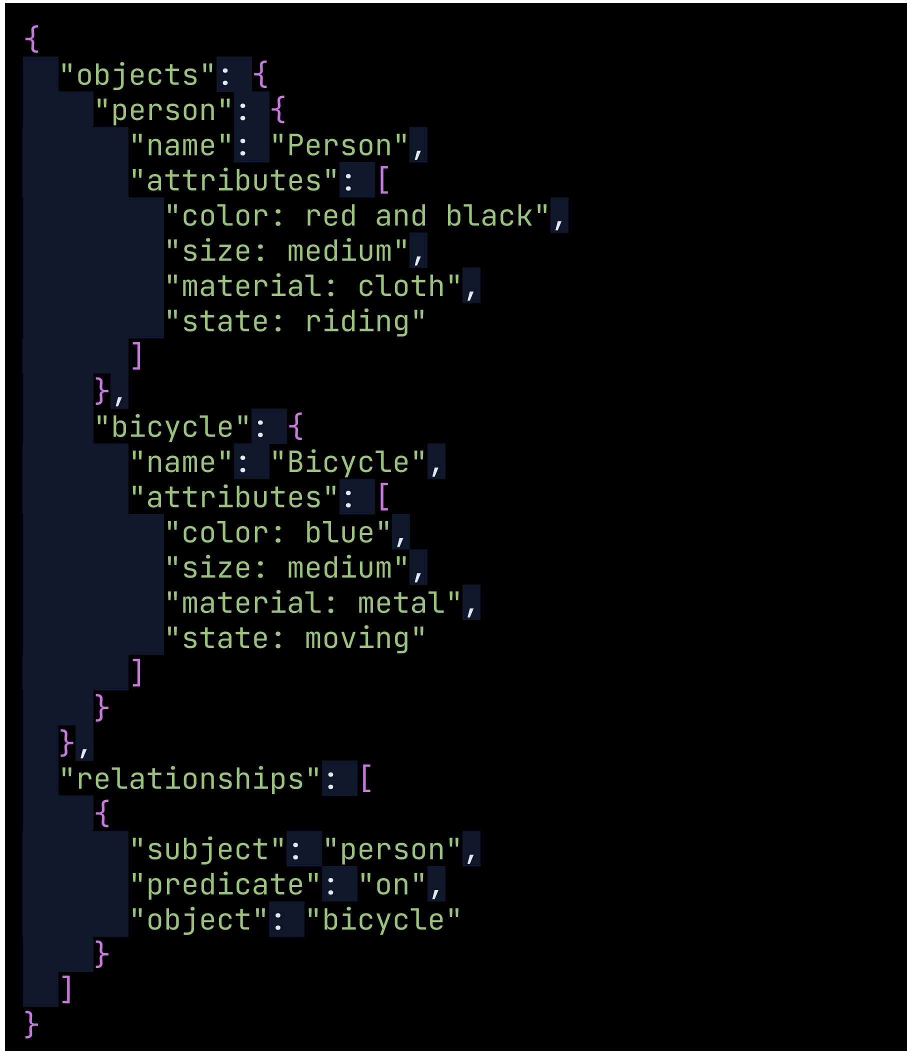
Phase 2: The Brain (LLaVA)

Scene Graph Construction

We employ **llava-v1.6-mistral-7b-hf** as the semantic engine. Its primary role is to connect the dots provided by DETR and enrich the scene understanding.

- **Relational Inference:** Identify edges (predicates) between DETR-detected objects (e.g., connect "cup" and "table" with "on").
- **Attribute Extraction:** Assign visual properties (color, material, state) to the detected entities.
- **Contextual Discovery:** Optionally identify and describe new objects or background elements missed by the fixed-set detector.

Example Scene Graph Output



```
{
  "objects": {
    "person": {
      "name": "Person",
      "attributes": [
        "color: red and black",
        "size: medium",
        "material: cloth",
        "state: riding"
      ]
    },
    "bicycle": {
      "name": "Bicycle",
      "attributes": [
        "color: blue",
        "size: medium",
        "material: metal",
        "state: moving"
      ]
    }
  },
  "relationships": [
    {
      "subject": "person",
      "predicate": "on",
      "object": "bicycle"
    }
  ]
}
```

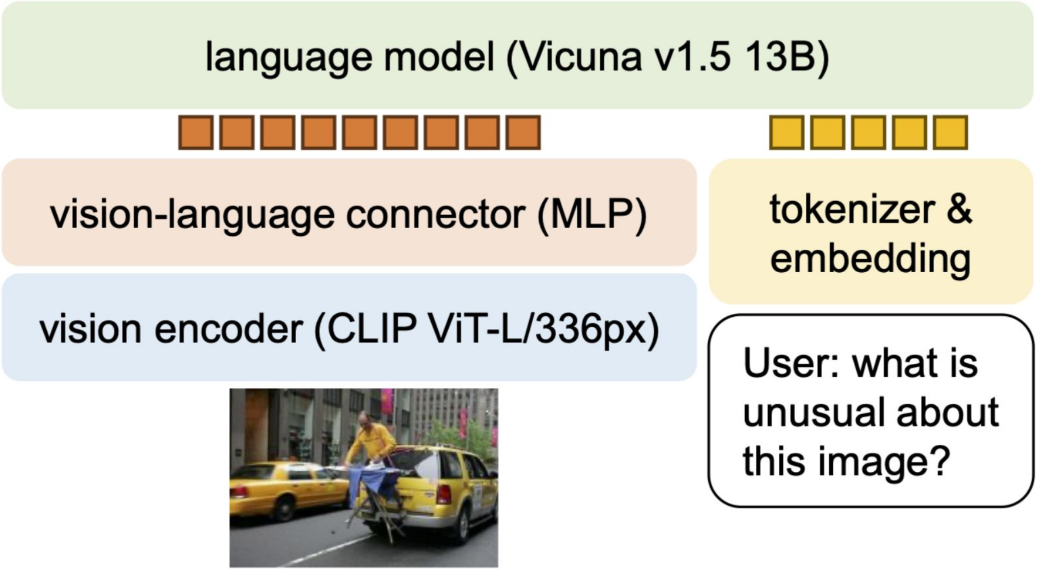
TECHNICAL DEEP DIVE

Model Focus: LLaVA v1.6 (Mistral)

Improved Visual Reasoning

LLaVA v1.6 is a CLIP-based vision-language model that connects a CLIP visual encoder with a large language model to enable image-aware conversational reasoning.

- **Dynamic Resolution:** Handles high-res images (up to 4x pixels) by breaking them into grids, preserving small details often lost in standard resize operations.
- **Architecture:** Connects a pre-trained CLIP ViT-L/14 visual encoder with the Mistral 7B LLM using a simple MLP projection layer. (Vicuna was used in earlier versions by Meta.)



language model (Vicuna v1.5 13B)
vision-language connector (MLP)
tokenizer & embedding
vision encoder (CLIP ViT-L/336px)
User: what is unusual about this image?

Source: Zhang et al., “Improved Baselines with Visual Instruction Tuning,” arXiv:2310.03744, 2023.

Context-Aware Prompting

Standard Prompting

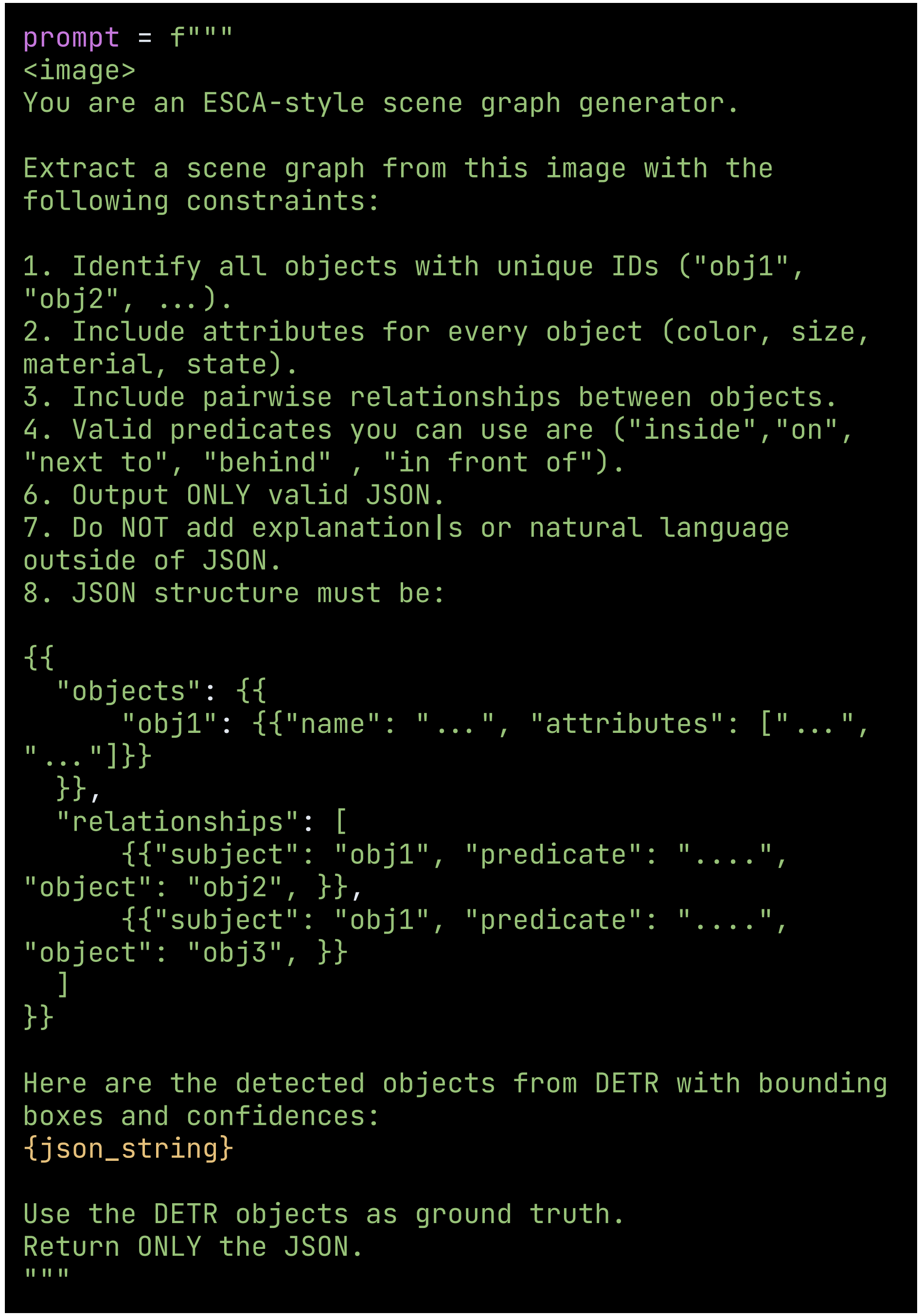
```
"Describe this image in detail."
```

Result: High hallucination risk. VLM invents plausible but non-existent objects.

✓ Context-Aware Injection

We **inject** the DETR ground truth directly into the prompt.

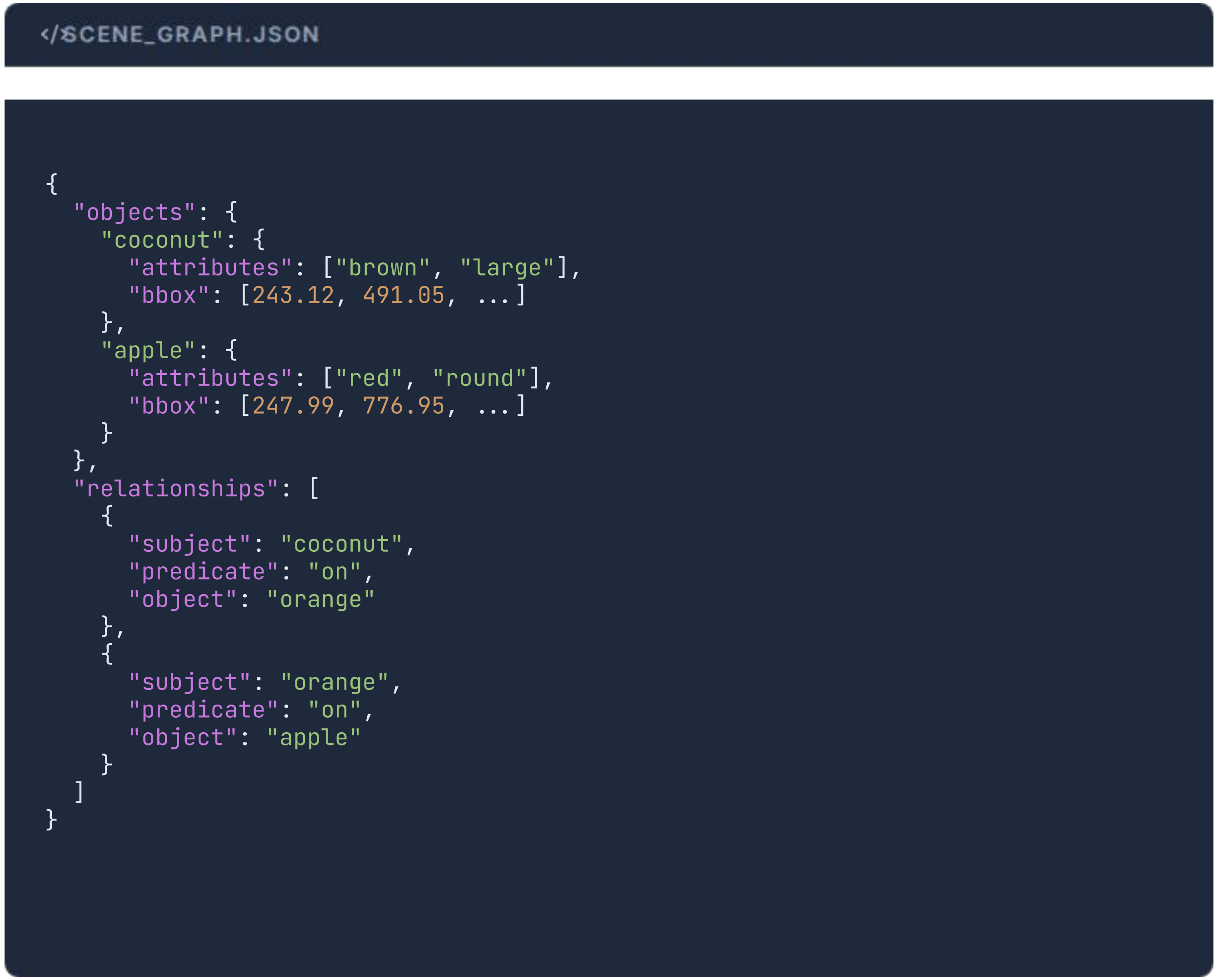
- **Constraint:** "Use DETR objects as ground truth."
- **Identity:** Assigns semantic names to specific bounding boxes.



```
prompt = f"""
<image>
You are an ESCA-style scene graph generator.

Extract a scene graph from this image with the
following constraints:

1. Identify all objects with unique IDs ("obj1",
"obj2", ...).
2. Include attributes for every object (color, size,
material, state).
3. Include pairwise relationships between objects.
4. Valid predicates you can use are ("inside","on",
"next to", "behind" , "in front of").
6. Output ONLY valid JSON.
7. Do NOT add explanation|s or natural language
outside of JSON.
8. JSON structure must be:

{{
  "objects": {{
      "obj1": {{"name": "...", "attributes": ["...",
"..."]}}
  }},
  "relationships": [
      {{"subject": "obj1", "predicate": "....",
"object": "obj2", }},
      {{"subject": "obj1", "predicate": "....",
"object": "obj3", }}
  ]
}}

Here are the detected objects from DETR with bounding
boxes and confidences:
{json_string}

Use the DETR objects as ground truth.
Return ONLY the JSON.
"""
```

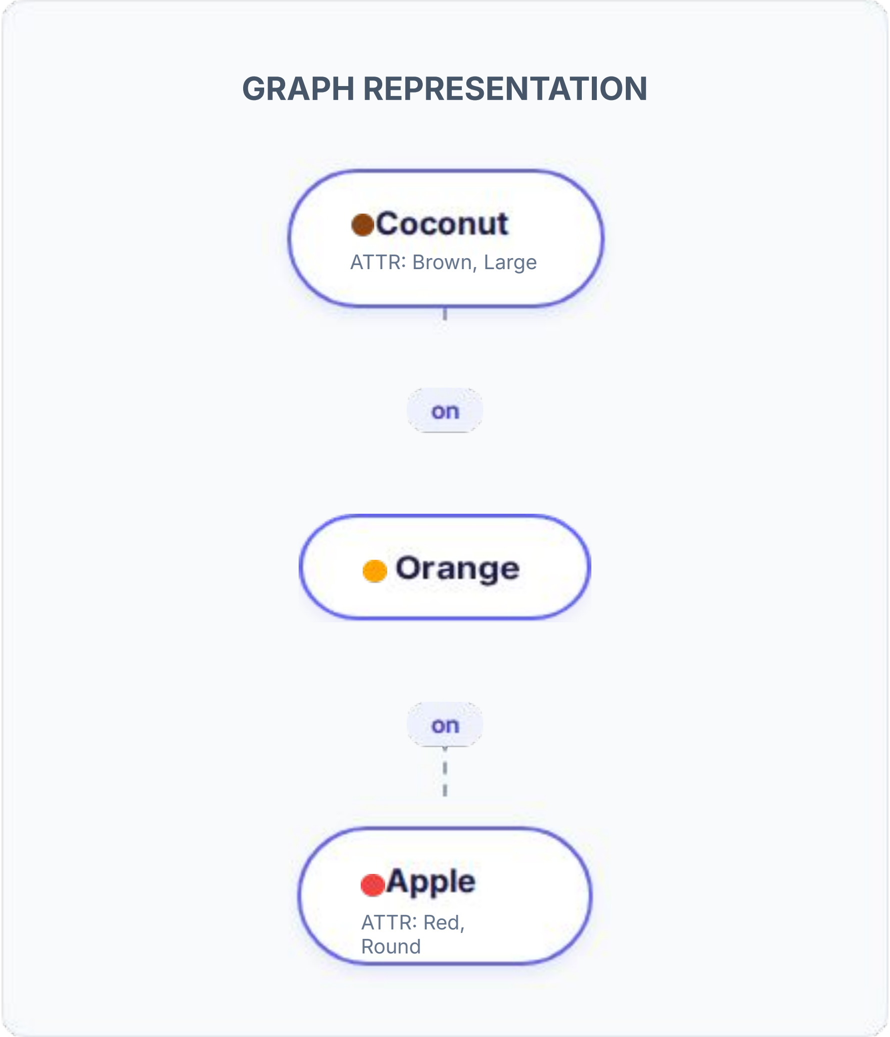
Resulting Scene Graph

The pipeline produces a structured, queryable JSON object that combines DETR's spatial precision with LLaVA's semantic richness.



</SCENE_GRAPH.JSON

```
{
  "objects": {
    "coconut": {
      "attributes": ["brown", "large"],
      "bbox": [243.12, 491.05, ...]
    },
    "apple": {
      "attributes": ["red", "round"],
      "bbox": [247.99, 776.95, ...]
    }
  },
  "relationships": [
    {
      "subject": "coconut",
      "predicate": "on",
      "object": "orange"
    },
    {
      "subject": "orange",
      "predicate": "on",
      "object": "apple"
    }
  ]
}
```



GRAPH REPRESENTATION

Coconut
ATTR: Brown, Large

on

Orange

on

Apple
ATTR: Red, Round

Symbolic Reasoning (CoT) Interface

Deriving answers from structured data using a frozen LLM (Mistral 7B).

USER INPUT

"Is coconut above the apple?"

CONTEXT (SCENE GRAPH)

```
{ "objects": ["coconut", "orange", "apple"], "relations": [
{"coconut": "on", "target": "orange"}, {"orange": "on",
"target": "apple"} ] }
```

MISTRAL 7B CHAIN-OF-THOUGHT

1 **Entity Identification**
Found object "coconut" and object "apple" in the graph.

2 **Path Traversal**
Detected path: coconut → on → orange → on → apple.

3 **Logical Inference**
Applied transitivity rule: If A is on B, and B is on C, then A is above C.

FINAL ANSWER

Yes, the coconut is above the apple indirectly.

Qualitative Result: Attribute Reasoning

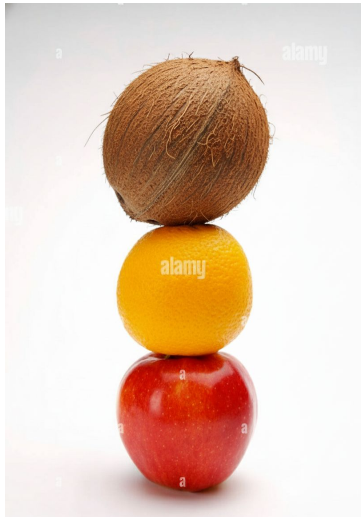
Demonstrating how the Scene Graph enables simple logical queries about object attributes.

USER QUERY

"Is the coconut above the apple?"

SYSTEM CONCLUSION

Yes, the coconut is above the apple indirectly.



ATTRIBUTE SEARCH TRACE

1. **Scan All Objects**
   Iterate over entities: coconut, orange, apple.
2. **Path Traversal**
   Detected path: coconut → on → orange
3. **Path Traversal continue**
   Detected path: orange→ on → apple
4. **Final Logic**
   Applied transitivity rule: If A is on B, and B is on C, then A is above C.

Comparison: Our Pipeline vs ESCA

Feature	Full ESCA (Paper)	Our Implementation
Concept Extraction	InternVL / GPT-4V	DETR
Localization	GroundingDINO + SAM2	DETR
Relation Inference	SGClip (Fine-tuned)	LLaVA + Verification
Compute Profile	Data Center (A100s)	Consumer (T4/L4)

Challenges & Limitations

Missing SGClip

Unavailability of SGClip forced reliance on zero-shot LLaVA for relation inference, reducing overall relation accuracy.

Closed-Set Limits

DETR is restricted to 80 COCO classes, unlike Open-Vocabulary detectors (GroundingDINO) that find arbitrary objects.

Model Scale

Smaller 7B models (Mistral/LLaVA) struggled with the complexity of real-world GQA images compared to larger models.

Prompting Inefficacy

CoT and Multi-Shot techniques yielded diminishing returns, as small 7B models lacked the capacity to fully utilize them.

Geometric Ambiguity

2D bounding box overlap cannot reliably distinguish complex depth relations like "behind" vs "inside".

Conclusion

- **Hybrid Efficiency:** We demonstrated that separating Localization (DETR) from Reasoning (LLaVA) significantly reduces hallucinations compared to end-to-end VLMs.

- **Geometric Grounding:** The "Judge" phase acts as a critical reliability layer, enforcing physical constraints on neural predictions.

- **Zero-Shot Viability:** The pipeline successfully performs visual reasoning tasks without task-specific training, leveraging pre-trained foundation models.

Q & A

Thank you for your attention.