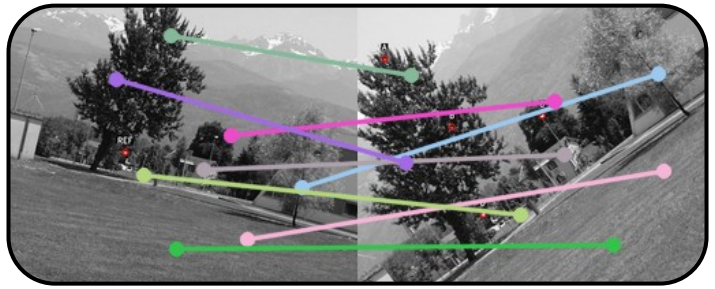
Don't Show Pixels, Show Cues: Unlocking Visual Tool Reasoning in Language Models via Perception Programs

Muhammad Kamran Janjua[1]* Hugo Silva[1]* Di Niu[2] Bahador Rashidi[1]
[1]Huawei Technologies, Canada [2]University of Alberta, Canada

Background

- Despite advancements in MLLMs, many vision problems are still highly benefitted by relying on external tools



Where did the points in the left image go?

In which direction is the camera moving?

Reconstruct this depth map

MLLM

❌ Prediction

✅ Ground truth

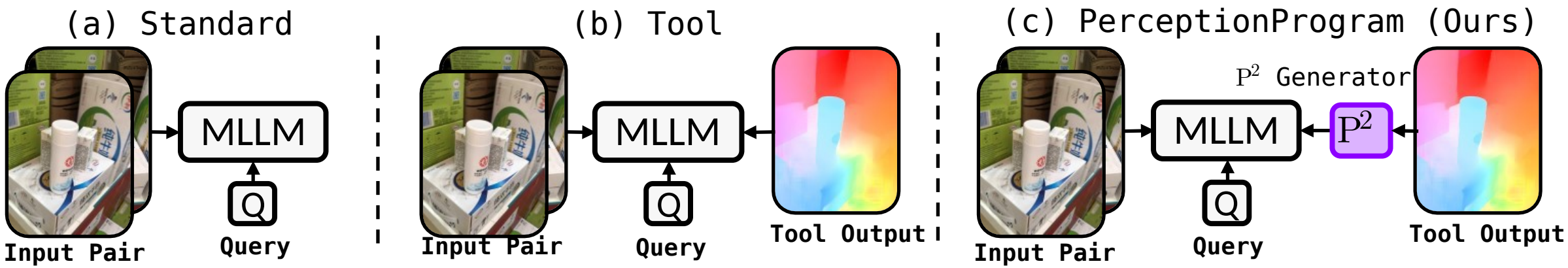
Background

- Main visual tool-calling approaches can be roughly classified in
  - Program synthesis
    - e.g. synthesize a script to compose vision tools
  - Chain-of-thought
    - e.g. emit tokens that carry vision information directly in the chain-of-thought
- Issues with existing approaches
  - Reliance on training and data-collection
  - Increased latency when composing multiple tool calls
  - Reasoning with pixels is harder than with text
- We ask the question

> *What if we could explain the expert outputs (depth map, optical flow, segmentation, etc.) to the model without it having to figure out what the modality means?*

Method

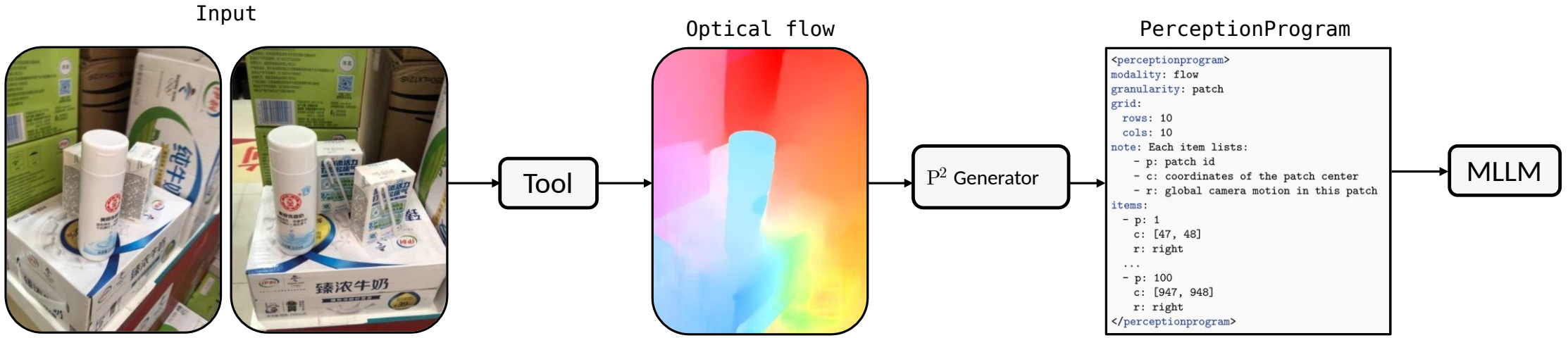
- PerceptionProgram
  - instead of relying on the MLLMs ability to interpret the pixel information directly, we preprocess it and convert it to a structured template



Example

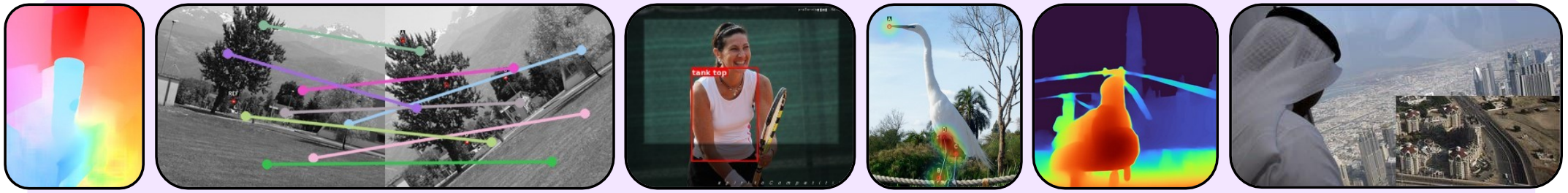
- The MLLM gets a textual description of two frames and then reasons about where the camera is moving

Multi-view reasoning



Method

- In the paper, we consider the following modalities



Multi-View Reasoning

```
<perceptionprogram>
modality: flow
granularity: patch
grid:
  rows: 10
  cols: 10
note: Each item lists:
    - p: patch id
    - c: coordinates of the patch center
    - r: global camera motion in this patch
items:
  - p: 1
    c: [47, 48]
    r: right
  ...
  - p: 100
    c: [947, 948]
    r: right
</perceptionprogram>
```

Visual Correspondence

```
<perceptionprogram>
modality: correspondence
granularity: keypoint
note: Each item lists one keypoint
     correspondence between two images:
    - p: running id
    - c: point in the first image
    - r: matched point in the second image
items:
  - p: 1
    c: [331, 16]
    r: [62, 164]
  ...
  - p: n // nth point
    c: [512, 961]
    r: [825, 899]
</perceptionprogram>
```

Object Localization

```
<perceptionprogram>
modality: object-detection
granularity: bounding-box
note: Each item lists a detected object:
    - p: detection id
    - c: bounding box coordinates [x0, y0,
         x1, y1]
    - r: detection confidence score (0-1)
    - b: object category label
items:
  - p: 1
    c: [285, 343, 578, 860]
    r: 0.7211
    b: tank top
</perceptionprogram>
```

Semantic Correspondence

```
<perceptionprogram>
modality: semantic
granularity: point
note: Each item lists a target candidate
     and its similarity:
    - p: target point
    - c: target point coordinate [x, y]
    - r: similarity score for each point
items:
  - p: A
    c: [137, 119]
    r: 0.268690
  ...
  - p: D
    c: [498, 726]
    r: 0.418051
</perceptionprogram>
```

Relative Depth

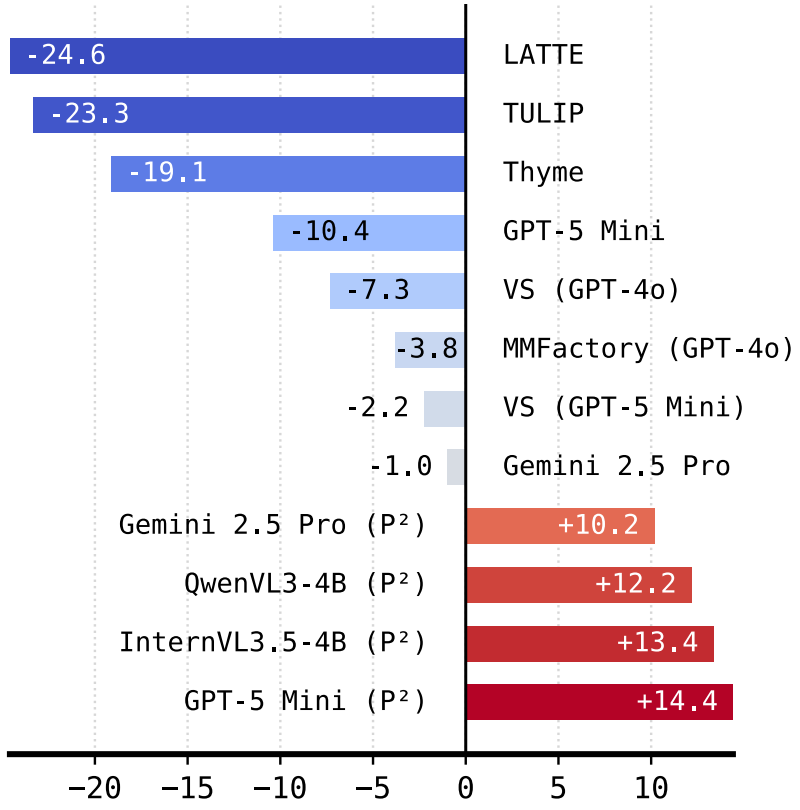
```
<perceptionprogram>
modality: depth
granularity: patch
grid:
  rows: 10
  cols: 10
note: Each item lists:
    - p: patch id
    - c: coordinates of patch center
    - r: min/max depth values for patch
items:
  - p: 1
    c: [47, 47]
    r: [0.3804, 0.4235]
  ...
  - p: 100
    c: [946, 946]
    r: [0, 0.02745]
</perceptionprogram>
```

Jigsaw

```
<perceptionprogram>
modality: jigsaw
granularity: border
note: Each item lists:
    - p: which border (image name)
    - c: candidate-piece strip bbox
    - r: average similarity score
items:
  - p: left (A)
    c: 0,0,30,1000
    r: 0.18849
  ...
  - p: top (B)
    c: 0,0,1000,55
    r: 0.38296
</perceptionprogram>
```

Main results

- Improves prior SoTA on BLINK tasks



- Works across architectures

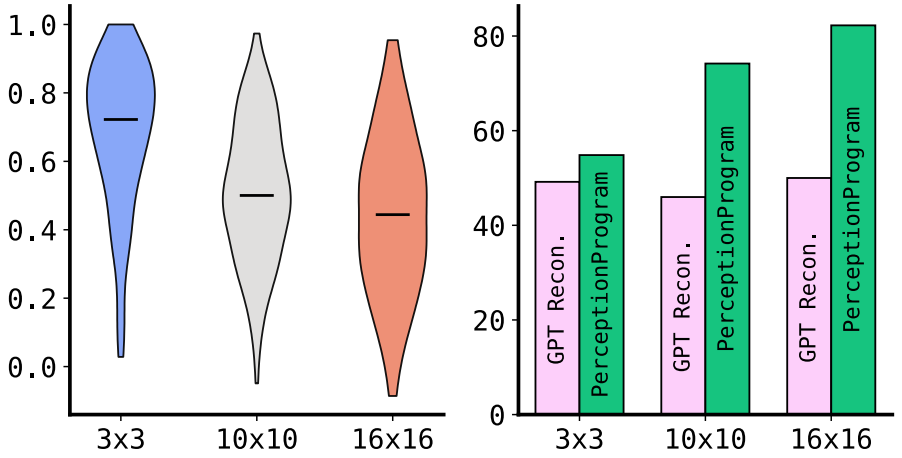
Model	Standard	Raw Tool	P^2 (Ours)
GPT-5 Mini (Zero-Shot)	59.66	60.67	**85.15**
Gemini 2.5 Pro (Zero-Shot)	68.38	66.52	**80.99**
Qwen3VL-4B (One-Shot)	60.17	59.23	**79.89**
InternVL3.5-2B (One-Shot)	42.79	44.96	**54.34**
InternVL3.5-4B (One-Shot)	55.67	57.30	**82.35**

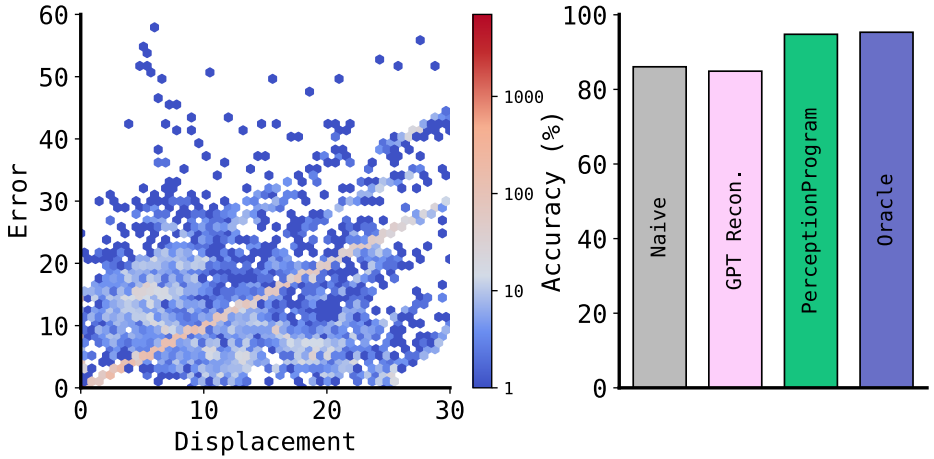
- Plugging P^2 directly other methods in significantly improves results

Methods	HardBLINK			Object Localization
	3	**4**	**5**	
VS	71.77	62.90	56.45	60.43
VS + P^2	**81.45**	**83.06**	**79.84**	**80.33**

Analysis

- When asked to do the Tool -> P^2 conversion themselves, MLLMs cannot interpret the visual information accurately





Conclusion

- The interpretation of pixel-level information in tool call pipelines hinders the ability of the model to reason. Structuring it and passing it as text is a simple and training-free technique that significantly enhances this capability across a wide range of model sizes.