Task Me Anything

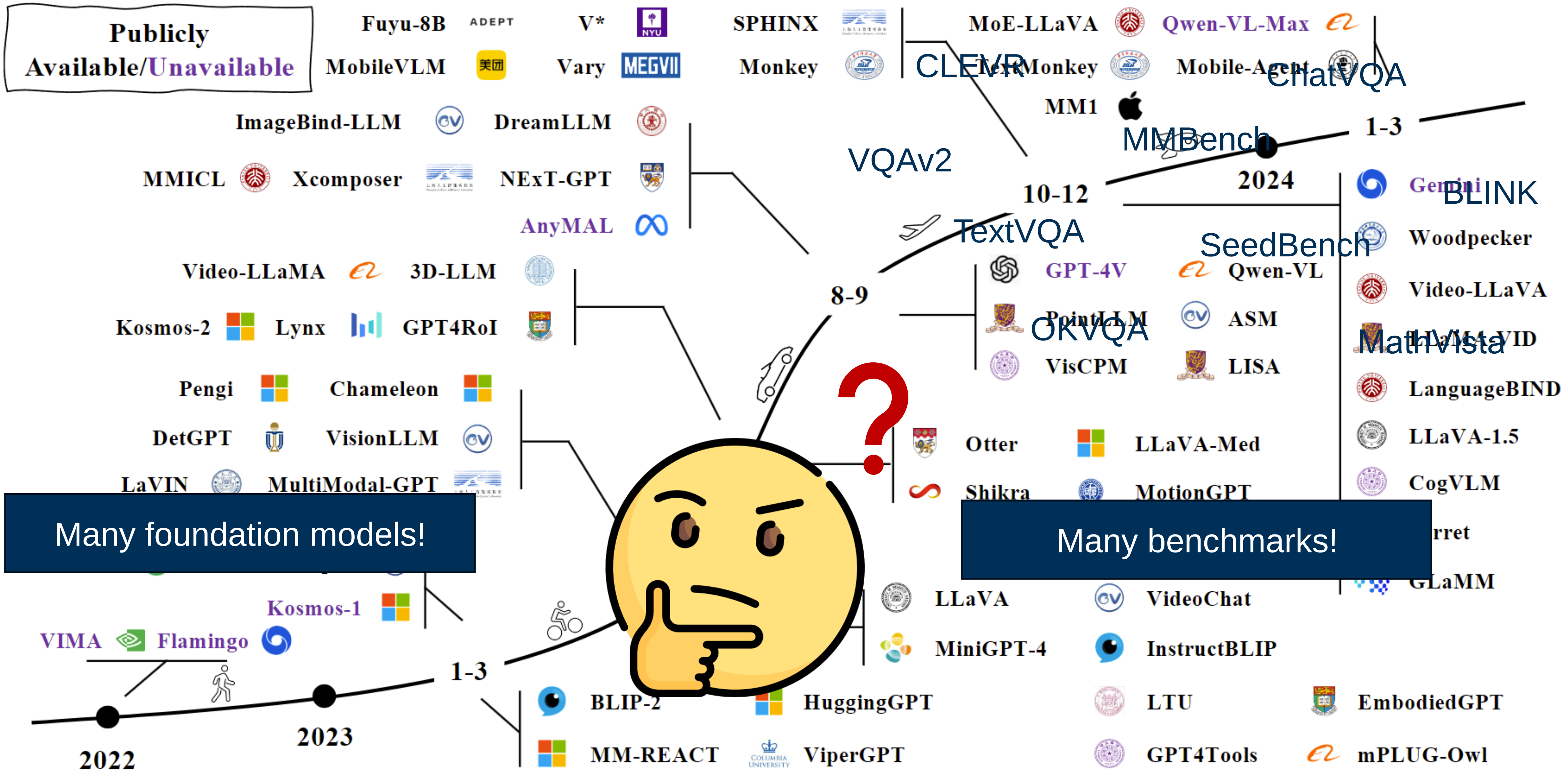
Jieyu Zhang, Weikai Huang, Zixian Ma, Oscar Michel, Dong He, Tanmay Gupta, Wei-Chiu Ma, Ali Farhadi, Aniruddha Kembhavi, Ranjay Krishna

Presented by:
Arthur Nascimento, Manisha Natarajan, Milad Heydari

Outline

- Problem Statement
- Background and Related Work
- Approach
- Experiments and Results
- Limitations , Societal Implications
- Summary of Strengths, Weaknesses, and relationship to other papers

Problem Statement



Many foundation models!

Many benchmarks!

Image Credit: A Survey on Multimodal Large Language Models, S. Yin et al.

Introduction

Need for user-centric benchmark generation:

(Q1): Which model is best at recognizing different plants?

(Q2): Which types of attributes is model X (say GPT4o) bad at recognizing?

Task Me Anything uses **procedural generation** to generate user-centric benchmarks for evaluating multimodal language models (MLMs)

(NO AI MODELS INVOLVED !!!)

Contributions:

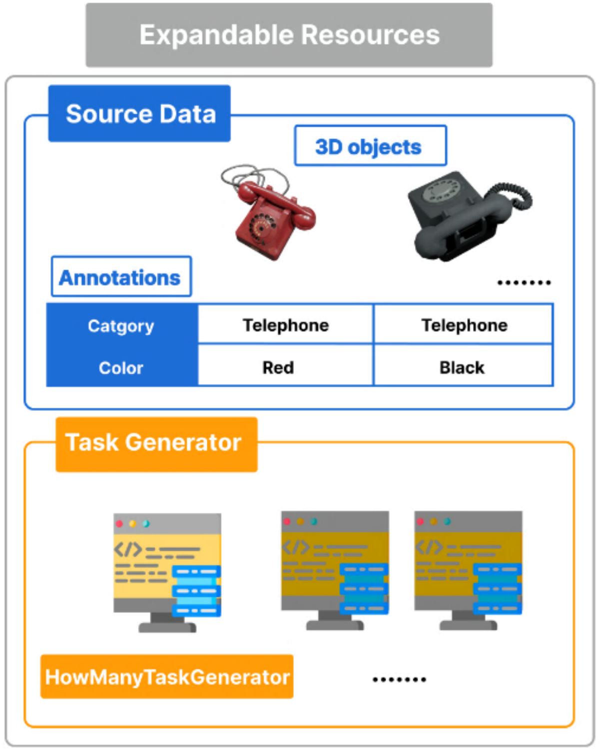
- ✅ User-centric benchmark generation
- ✅ Expandable Task space (~ 750M tasks)
- ✅ Supports fine-grained user query with budget approximation

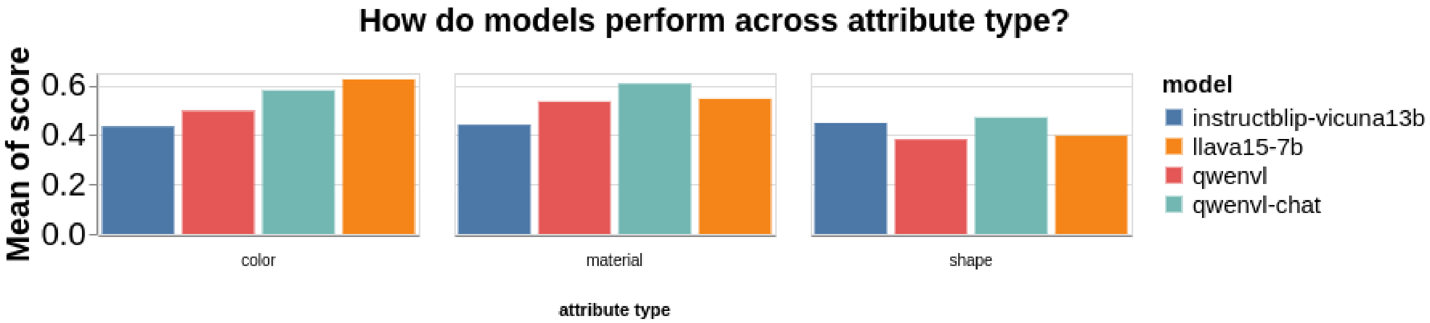
Introduction

Two key components for benchmark generation engines:

Generation

Evaluation





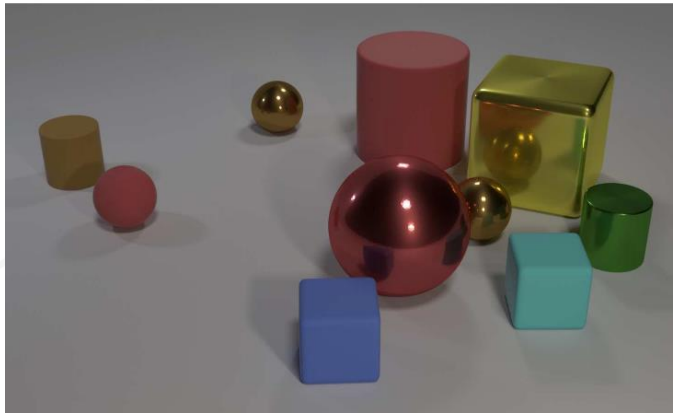
Related Work

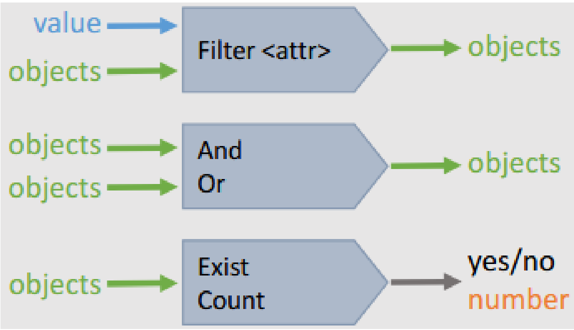
Programmatic Task Generation

Leveraging scene graphs:

CLEVR Dataset [1]:

Three shapes, two sizes, two materials and eight colors, four relations





Q: How many red things are there?

↓

“How many <C> <M> things are there?”

↓

count(filter color(<C>, filter material(<M>, scene())))

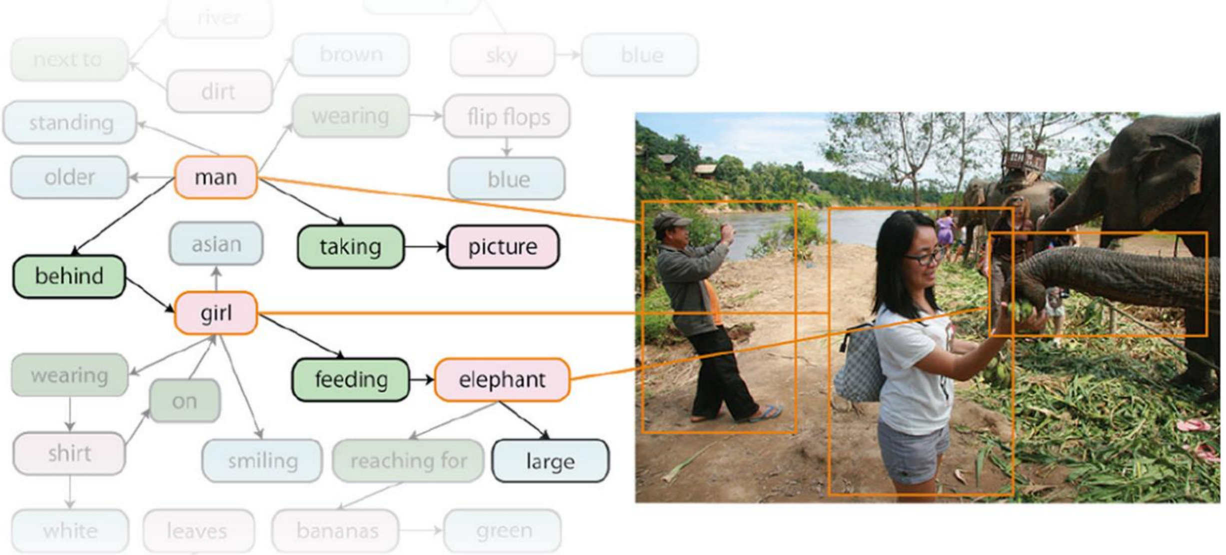


Image Credit: Visual genome: Connecting language and vision using crowdsourced dense image annotations. International journal of computer vision, R. Krishna et al.

Image Generation: Sample scene graph, use Blender

Question and Answer Generation:

Pick question family, fill in template values (avoid ill-posed / degenerate values with DFS), rejection sampling

Use pre-defined programs (query object attributes, compare, count)

[1] Johnson, J., Hariharan, B., Van Der Maaten, L., Fei-Fei, L., Lawrence Zitnick, C. and Girshick, R., 2017. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2901-2910).

Programmatic Task Generation

Question Pattern:

What <type> is <Object>, <attribute> or <Attribute>?

What color is the apple on the white table, red or green?

select: table → filter: white → relate(subject,on): apple → query: color

Leveraging Scene Graphs:

GQA Dataset [1]:

Similar scene graph-based approach to VQA for more realistic scenes

Pattern: What|Which <type> [do you think] <is> <dobject>, <attr> or <decoy>?

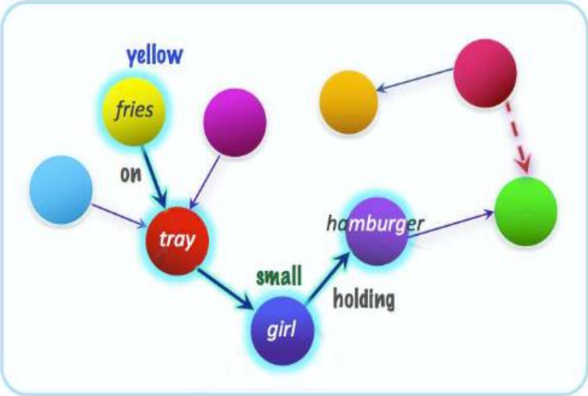
Program: **Select:** <dobject> → **Choose** <type>: <attr>|<decoy>

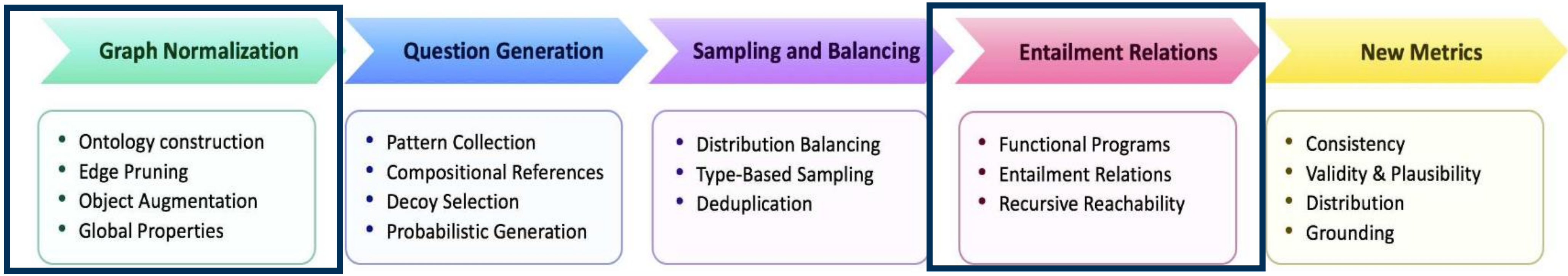
Reference: The food on the red object left of the small girl that is holding a hamburger

Decoy: brown

What color is the food on the red object left of the small girl that is holding a hamburger, yellow or brown?

Select: hamburger → **Relate:** girl,holding → **Filter** size: small → **Relate:** object, left → **Filter** color: red → **Relate:** food,on → **Choose** color: yellow | brown





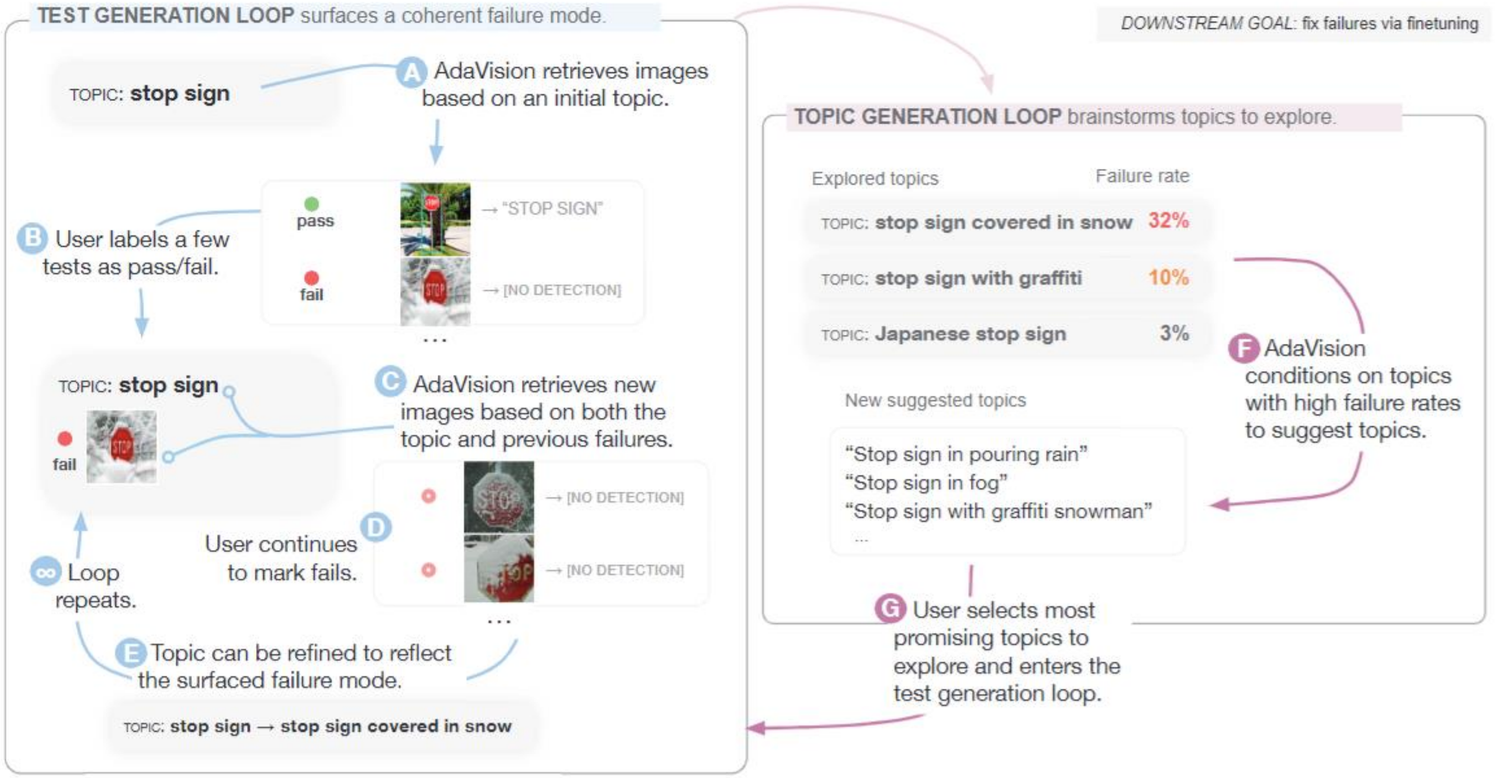
[1] Hudson, D.A. and Manning, C.D., 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 6700-6709).

Adaptive Evaluation

Train Set	Validation	Test

Adaptive Testing and Debugging:

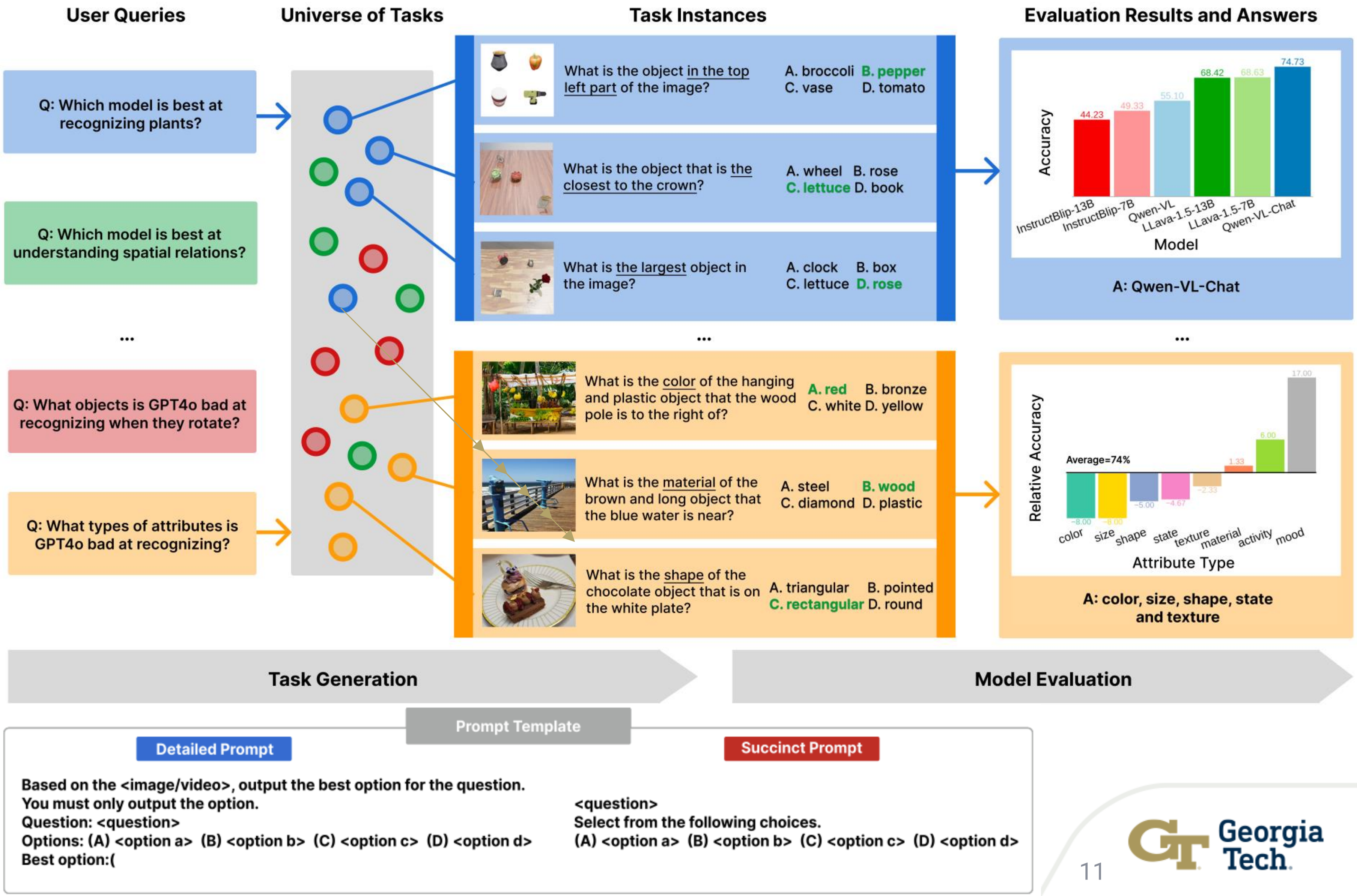
- Dynamically update test data (e.g., Dynabench, LatestEval)
- Adaptively identify task groups where the model underperforms



Approach

Approach: Overview

- User brings benchmarking queries
- Program Identify relevant tasks
- Generates relevant VQA Task Instances
- Evaluate model on Scene Graphs & Generated Images
- Expandable processes



Georgia Tech

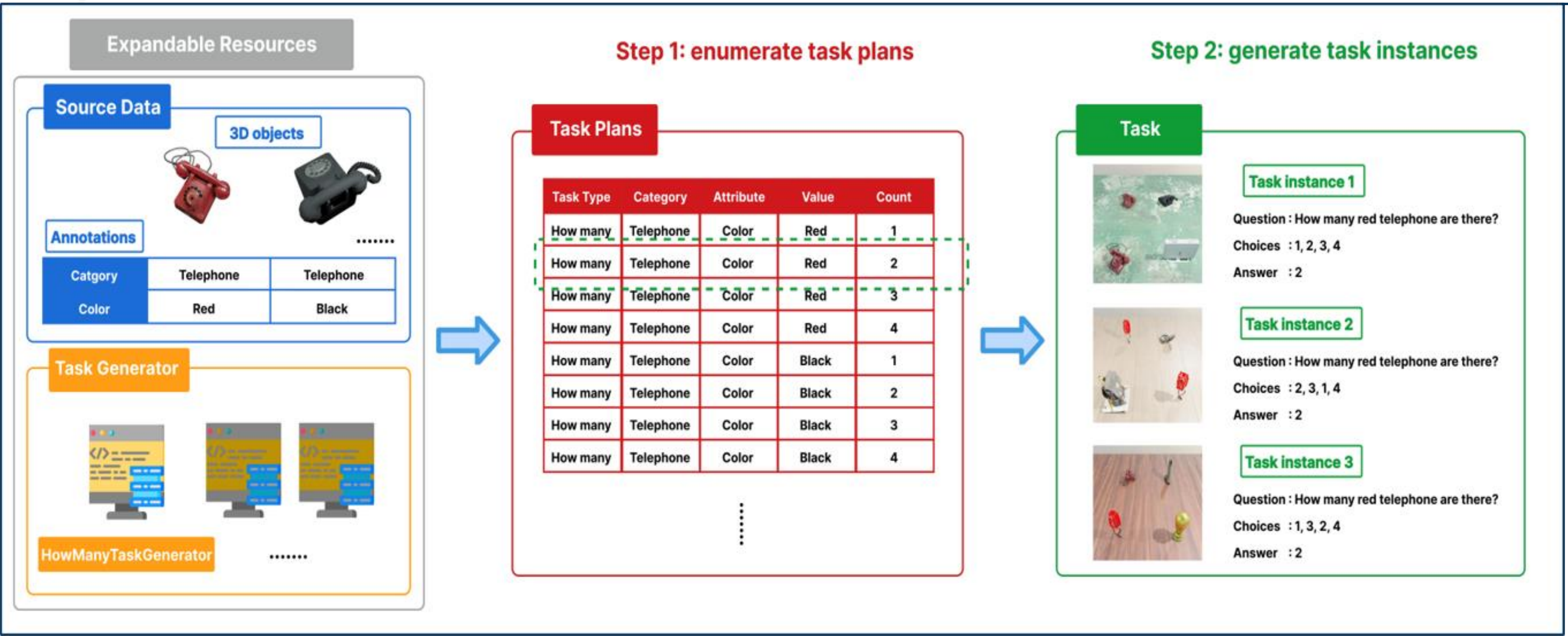
Approach: Terminology

Example :

- **Source Data:** Images, scenes, attributes
- **Query:** What model is best at counting red Telephones?

Task generator (program):

1. Uses the benchmarking **query** (input) to enumerate all relevant **task plans** (table)
2. Finds relevant **tasks** (questions + mc)
3. Generates **task instances** (Images) for VQA



Taxonomy: Task Space / Assets

Task Generator: program

Task Plans: meta data

Task : The question + MC

Task Instance: VQA + Image

Approach: Generating Tasks

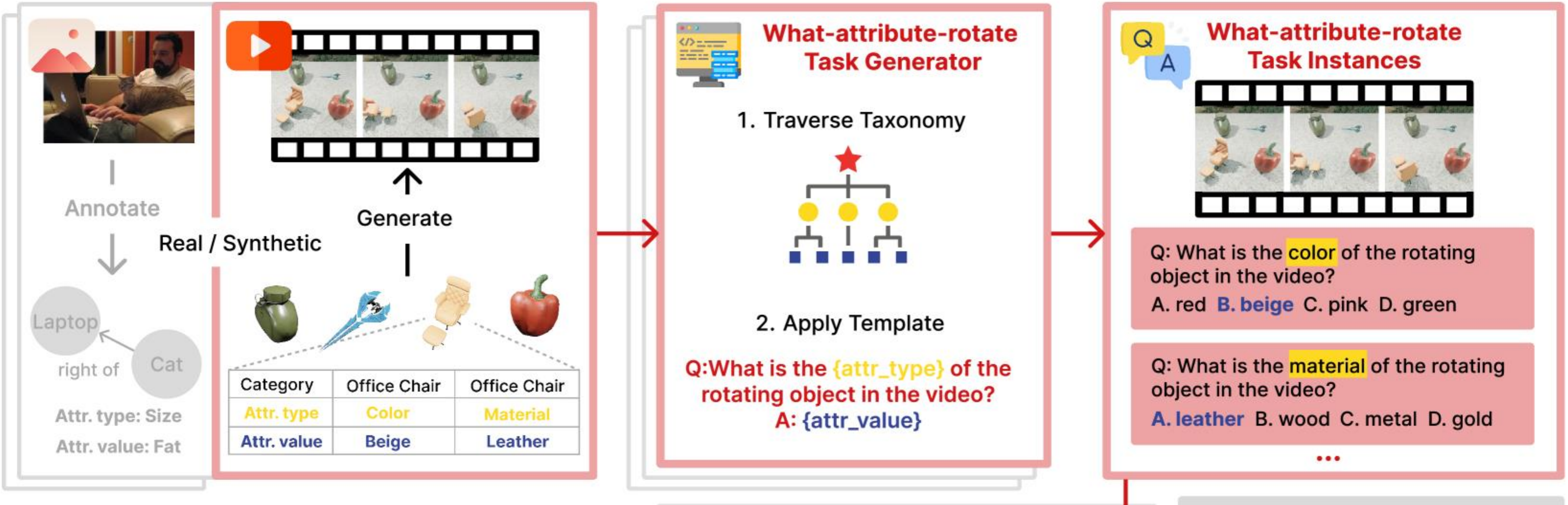
- Images/videos programmatically rendered using Blender / image renderer

  OR

- Real images used with pre-annotated scene graph

- Questions programmatically generated from templates + taxonomy/attributes

- False but plausible answers generated using Adversarial Filtering (LLM, but no image input)



Approach: Scene Types

Skills

- counting
- spatial
- relation
- attribute
- 3D attribute
- object
- temp. attr
- action

Image QA

Visual Input 1: 2D Sticker Image

Task Generator	Number of Tasks
how many	32,487
what	58,689,137
where	58,689,137
what attribute	47,541,884
where attribute	47,541,884
Total	212,494,529

Visual Input 2: 3D Tabletop Scene

Task Generator	Number of Tasks	Task Generator	Number of Tasks
how many	32,487	what size	19,688
what	58,689,137	what attribute size	15,968
where	58,689,137	where size	54,164,744
what attribute	47,541,884	what distance	58,657,144
where attribute	47,541,884	what attribute distance	47,515,936
		where distance	58,657,144
Total			431,525,153

Visual Input 3: Scene Graph with Real Image

Task Generator	Number of Tasks
what object	25,173
what attribute	20,546
what relation	23,241
Total	68,960

Video QA

Visual Input 4: 3D Tabletop Scene

Task Generator	Number of Tasks
what rotate video	39,376
what attribute rotate video	31,936
where rotate video	108,329,488
what move video	78,752
what attribute move video	63,872
where move video	78,752
Total	108,622,176

Visual Input 5: Scene Graph with Real Video

Task Generator	Number of Tasks
what object video	428,342
what relation video	428,342
what action video	335,386
Total	1,192,070

Georgia Tech

Approach: Task Space

- **133k** Images, 10k videos, 2k 3D objects
- **365** Object Categories
- **655** Attributes (color, texture, size)
- **335** Relationships (spatial, modeled within scene graphs)
- **28** Task Generators (how many, what color)
- **5** Types of Visual Input (2D tabletop, 3D, video)
- **750M** possible image/video question-answering pairs
- Much larger than comparable datasets: GQA(22M), (CLEVR 100k)
- Can be specialized for specific tasks

Georgia Tech

Approach: Query Types

Top-K

Find Top-5 objects that GPT-4o is worst at recognizing when rotating

Threshold

Identify colors that GPT-4o recognizes with less than 30% average accuracy

Model Compare

Compare with LLaVA-Next to determine which objects GPT-4o performs better at recognizing

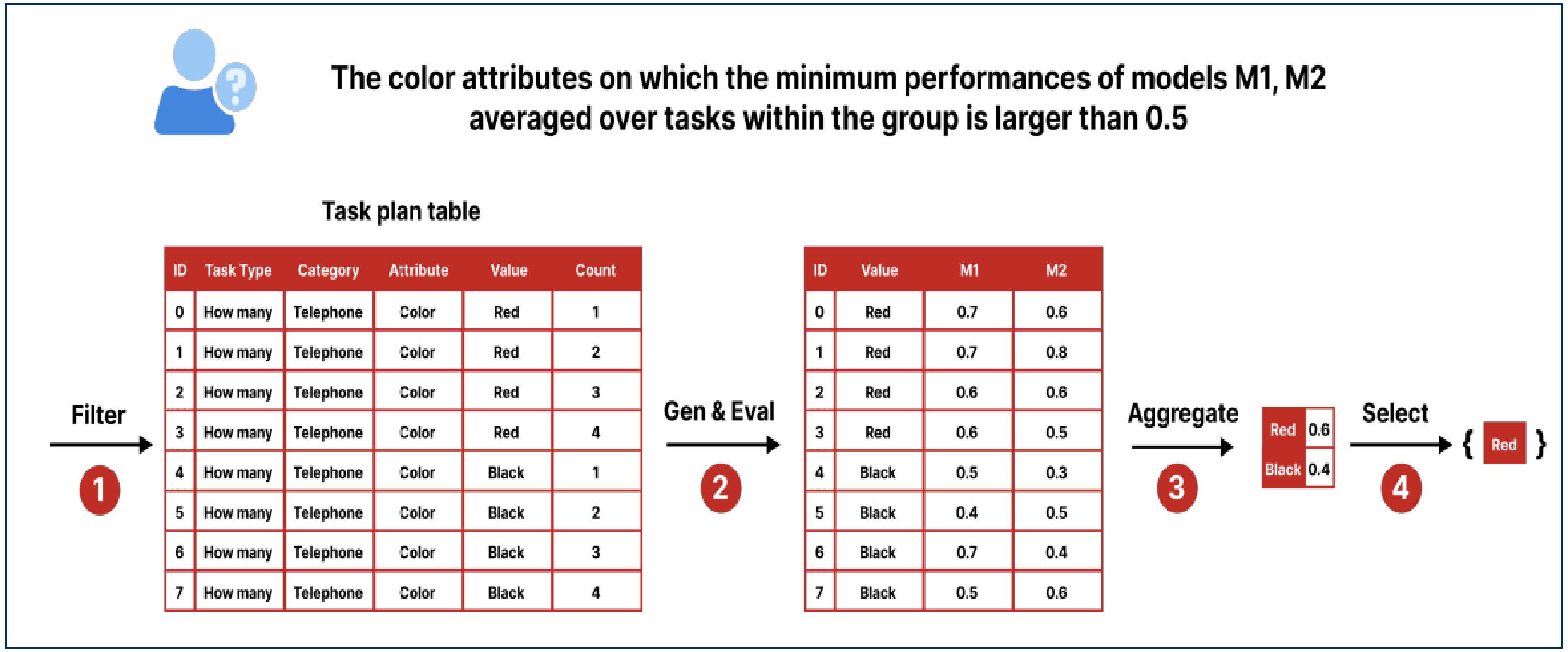
Model Debug

Identify tasks for which InstructBLIP performs significantly worse than its average performance

- Querys are not open language.
- Must be written 'SQL' style language using the set of attributes, query types, relationships, etc.

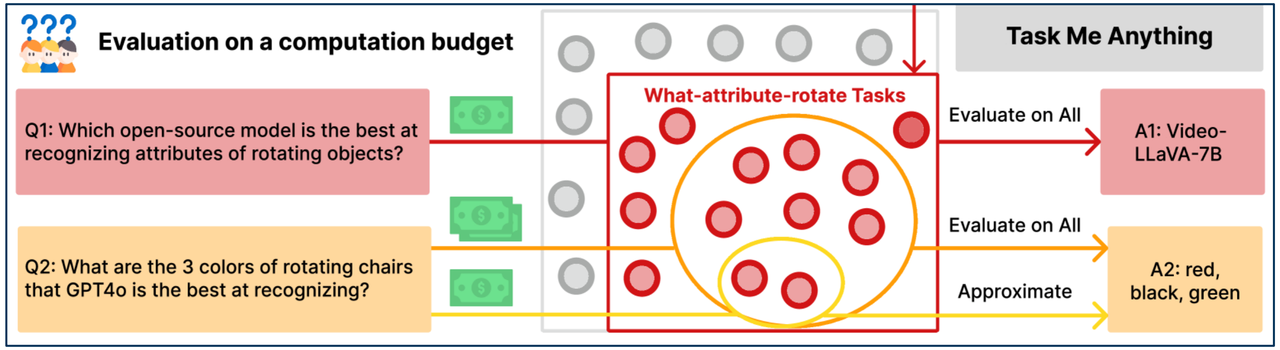
Approach: Query Execution / Cost Optimization

- **Random Sampling**: Randomly selects a subset of task instances to evaluate MLMs.
- **Fitting**: Trains regressor to predict MLM performance based on past samples and task metadata.
- **Active Learning**: Iteratively refines the regressor by sampling most uncertain task instances for improved predictions.



ID	Task Type	Category	Attribute	Value	Count
0	How many	Telephone	Color	Red	1
1	How many	Telephone	Color	Red	2
2	How many	Telephone	Color	Red	3
3	How many	Telephone	Color	Red	4
4	How many	Telephone	Color	Black	1
5	How many	Telephone	Color	Black	2
6	How many	Telephone	Color	Black	3
7	How many	Telephone	Color	Black	4

ID	Value	M1	M2
0	Red	0.7	0.6
1	Red	0.7	0.8
2	Red	0.6	0.6
3	Red	0.6	0.5
4	Black	0.5	0.3
5	Black	0.4	0.5
6	Black	0.7	0.4
7	Black	0.5	0.6



Approach: Output/Contributions

- **TASK-ME-ANYTHING:** Task Generator process/code itself. Expandable with new datasets/features and can be used to generate new, custom benchmarks
- **TASK-ME-ANYTHING-RANDOM**: 100 random tasks, 5700 ImageQA and 2700 VideoQA instances. Evaluated 18 MLMs with detailed and succinct prompts
- **TASK-ME-ANYTHING-DB**: Over 100K tasks generating 1M+ instances. 13 MLMs evaluated across 24.24 million evaluation pairs. Results aid in model performance prediction
- **TASK-ME-ANYTHING-UI**: Graphical interface with tabs for model performance, task embedding visualization, performance anomalies, and detailed query investigations

Georgia Tech

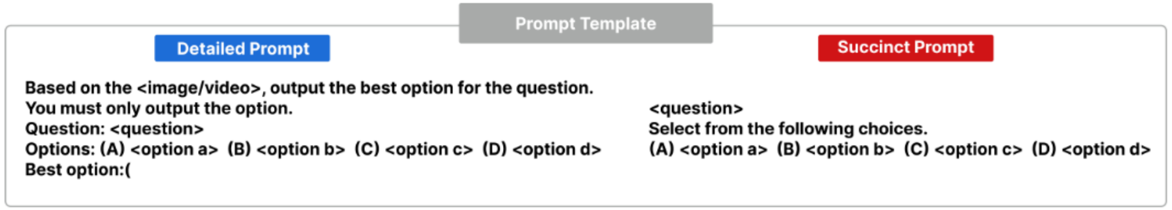
Experiments and Results

Experiments and Results: Overview

Pipeline for automatic benchmark generation paper

Benchmarking of previous work!

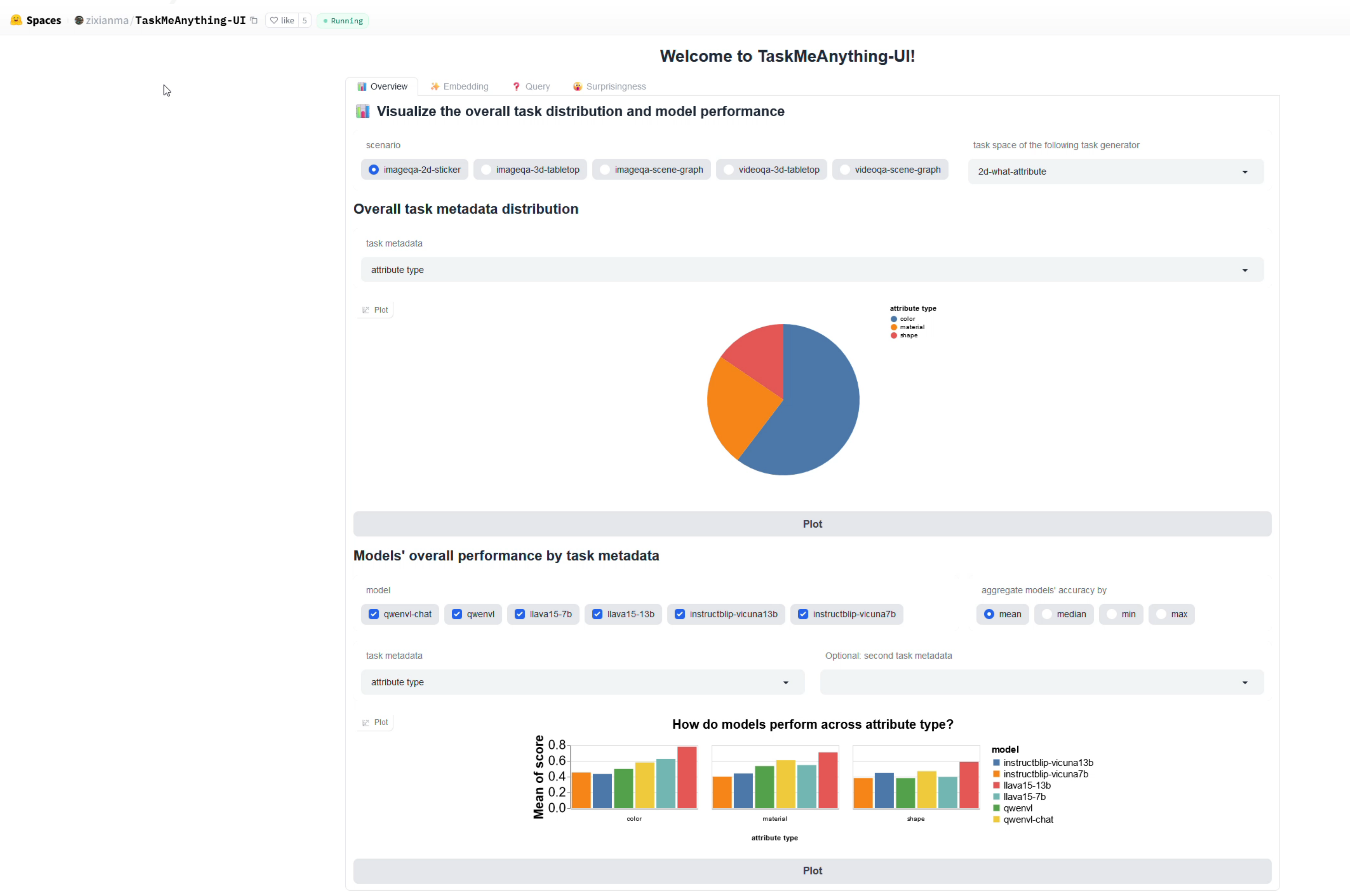
- Evaluation of 18 different VLMs across 8400 tasks instances made public



100 random tasks
3 instances each

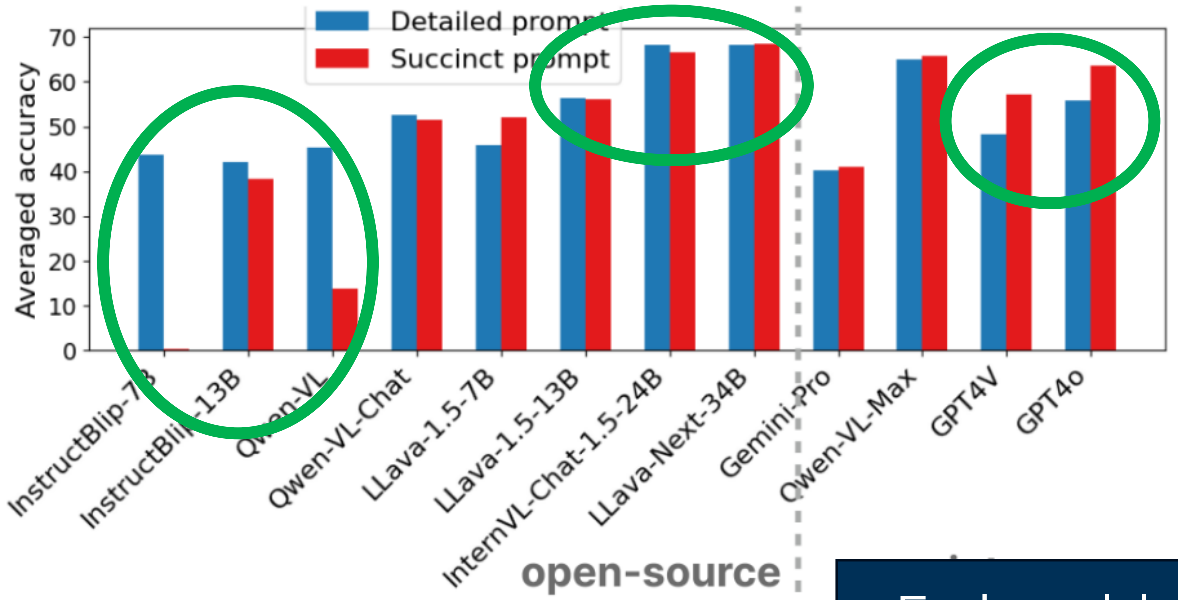
100k random tasks
15 instances each

1) Proposed benchmark: random subset of 2700 (IQA)/5700 (VQA) tasks instances (TMA-Random)
2) Open source VLMs (13): over 1M tasks instances (TMA-DB)
3) UI to explore TMA-DB with different queries (TMA-UI)

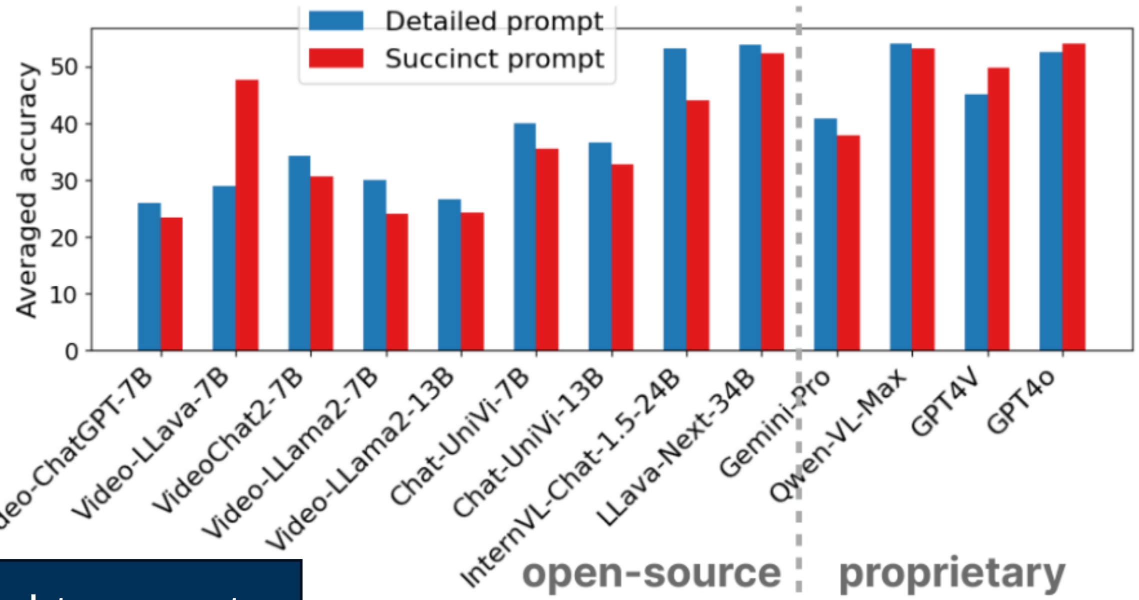

Spaces
zixianma / TaskMeAnything-UI
like 5
Running
App
Files
Community
Welcome to TaskMeAnything-UI!
Overview
Embedding
Query
Surprisingness
Visualize the overall task distribution and model performance
scenario
imageqa-2d-sticker
imageqa-3d-tabletop
imageqa-scene-graph
videoqa-3d-tabletop
videoqa-scene-graph
task space of the following task generator
2d-what-attribute
Overall task metadata distribution
task metadata
attribute type
Plot
attribute type
color
material
shape
Plot
Models' overall performance by task metadata
model
qwenvl-chat
qwenvl
llava15-7b
llava15-13b
instructblip-vicuna13b
instructblip-vicuna7b
aggregate models' accuracy by
mean
median
min
max
task metadata
attribute type
Optional: second task metadata
Plot
How do models perform across attribute type?
Mean of score
0.8
0.6
0.4
0.2
0.0
color
material
shape
attribute type
model
instructblip-vicuna13b
instructblip-vicuna7b
llava15-13b
llava15-7b
qwenvl
qwenvl-chat
Plot

Q1: How do models perform over a random subset of all possible questions?

ImageQA



VideoQA



Each model might present different prompt sensitivity

Detailed Pro

Based on the <image/video>, output the best option for the question.
You must only output the option.
Question: <question>
Options: (A) <option a> (B) <option b> (C) <option c> (D) <option d>
Best option:(

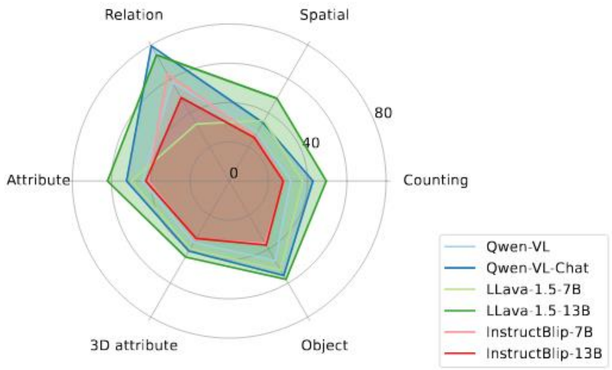
t Prompt

<question>
Select from the following choices.
(A) <option a> (B) <option b> (C) <option c> (D) <option d>

Georgia Tech

Q2: What is the best MLM for each specific skill? (IQA)

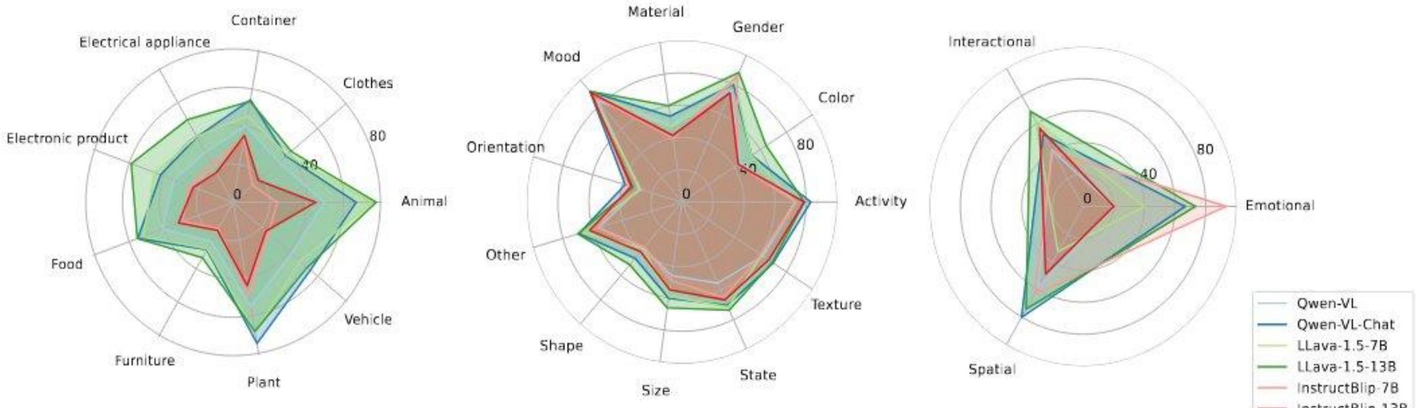
High-level skills



(a) ImageQA

Different models have different expertise

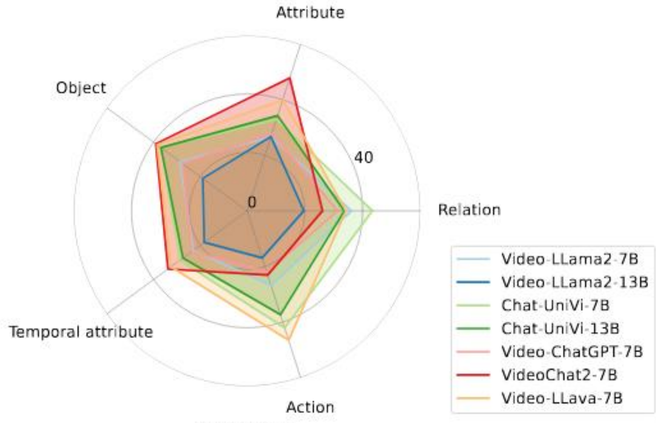
Fine-grained



(a) Object (b) Attribute (c) Relation

Q3: What is the best MLM for each specific skill? (VQA)

High-level skills

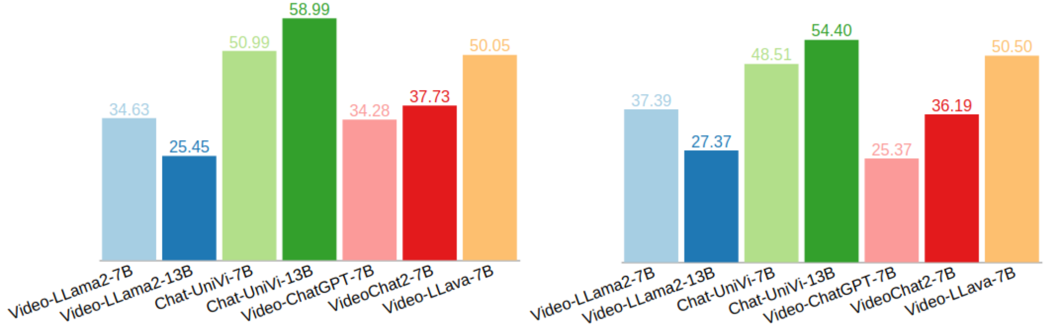


(b) VideoQA

In VQA, bigger difference in expertise per model

Fine-grained

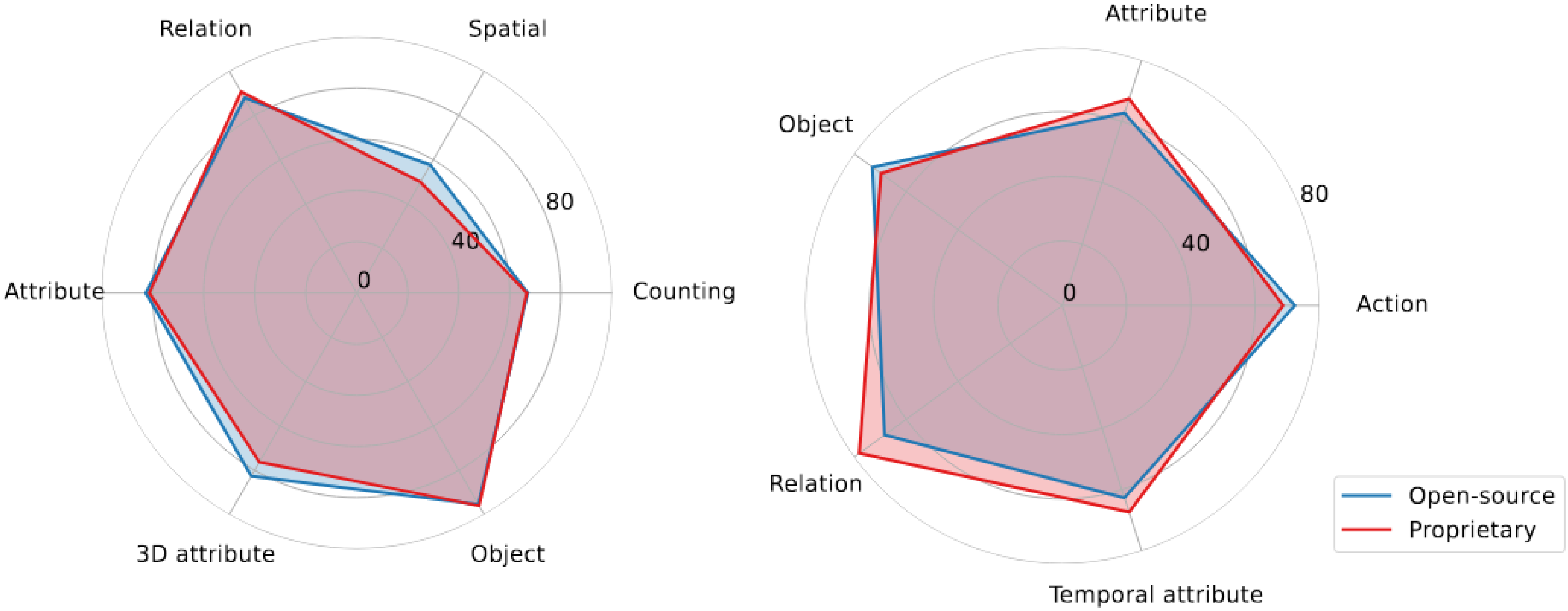
(a) Object (b) Attribute



(a) Spatial **(b) Interactional**

Q4: How does the best open-source model compare against the best proprietary model across skills?

open source VS



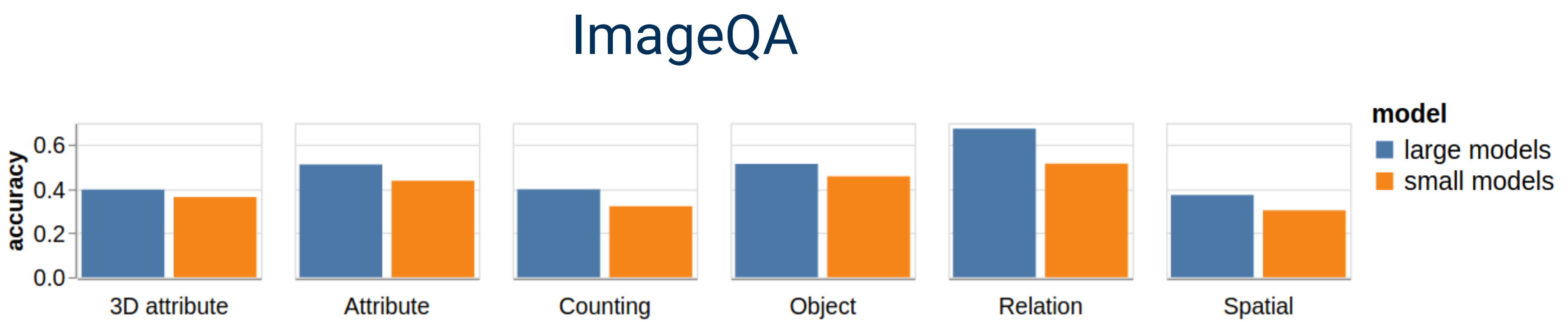
(a) ImageQA

(b) VideoQA

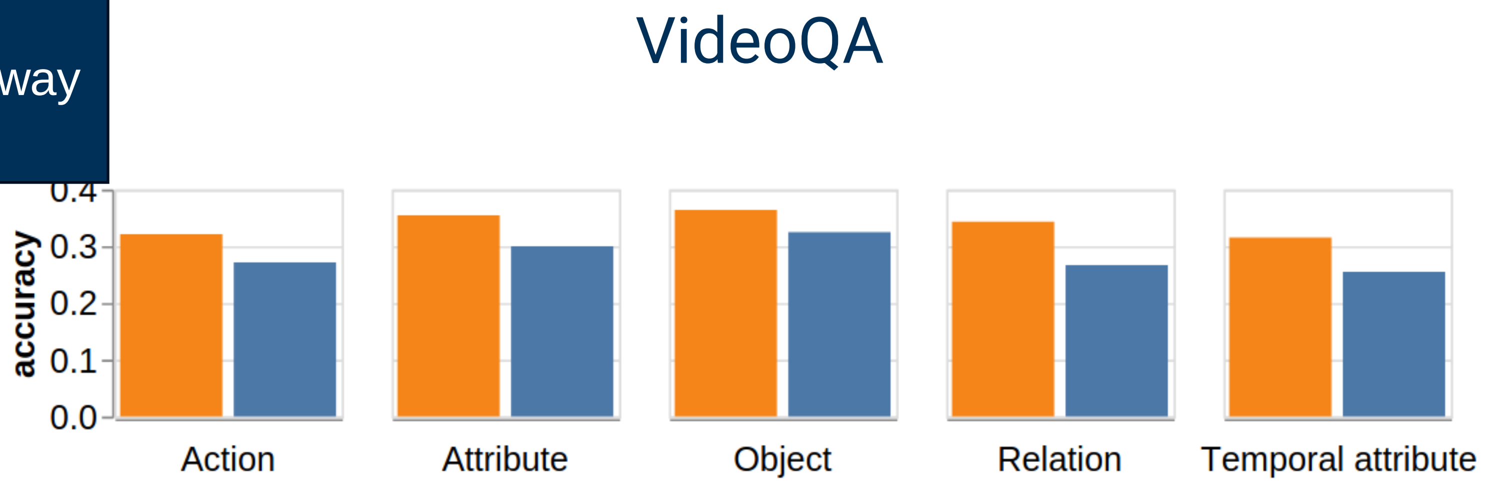
Open and closed source models are overall competitive

Georgia Tech

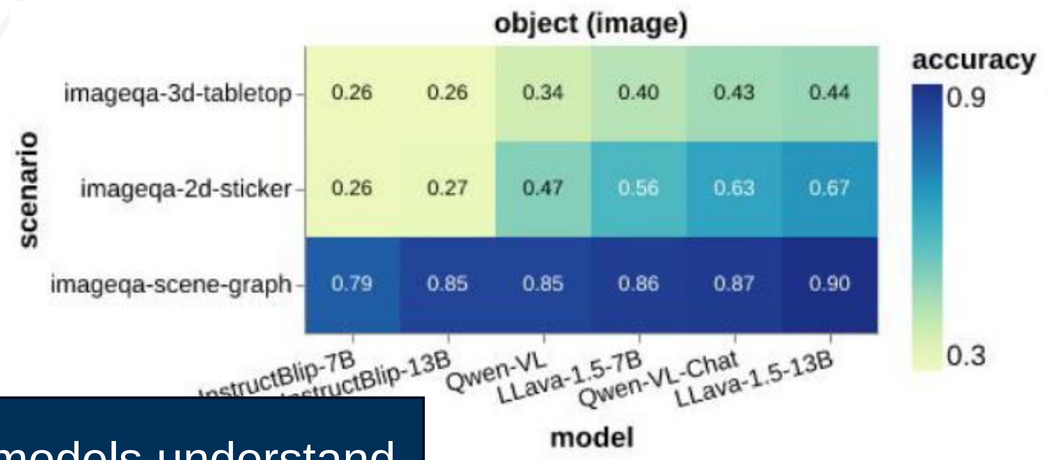
Q5: How do small models compare against large models?



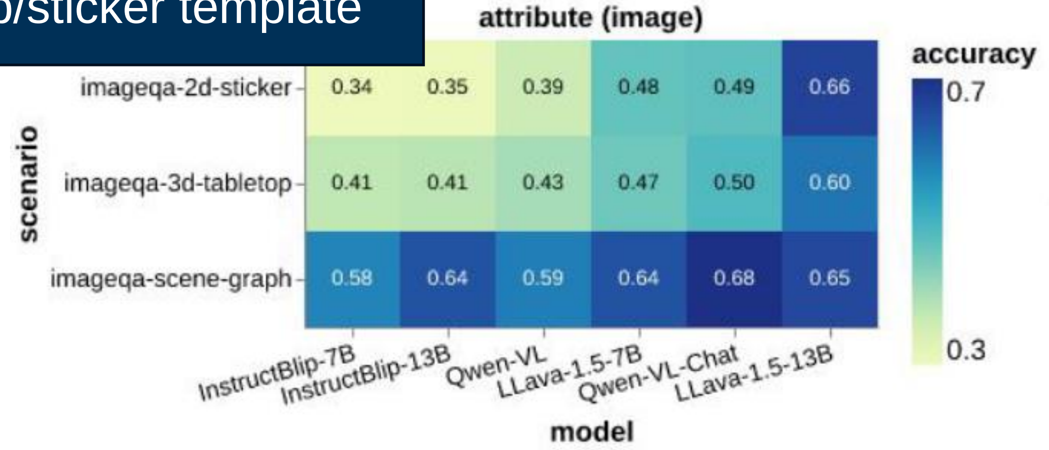
ImageQA: the larger the better (overall)
VideoQA: the other way around (!!!)

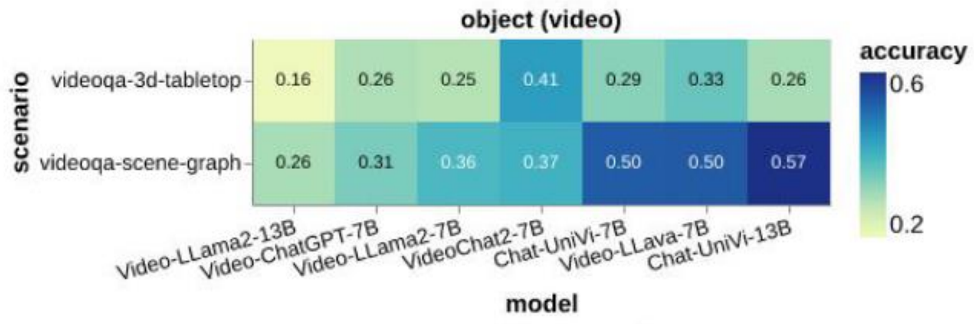


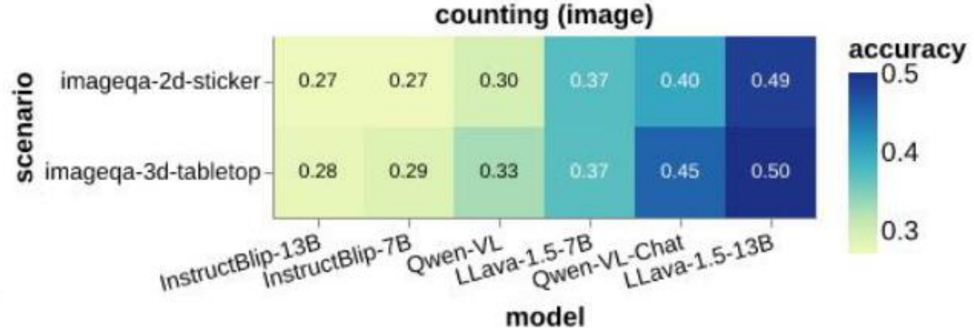
Q6: Are models' strengths and weaknesses consistent across visual inputs?

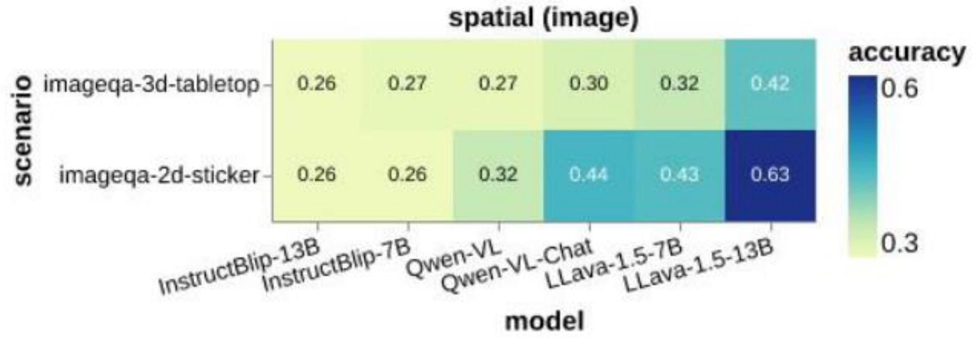


Overall, models understand scene-graph better than 3D tabletop/sticker template









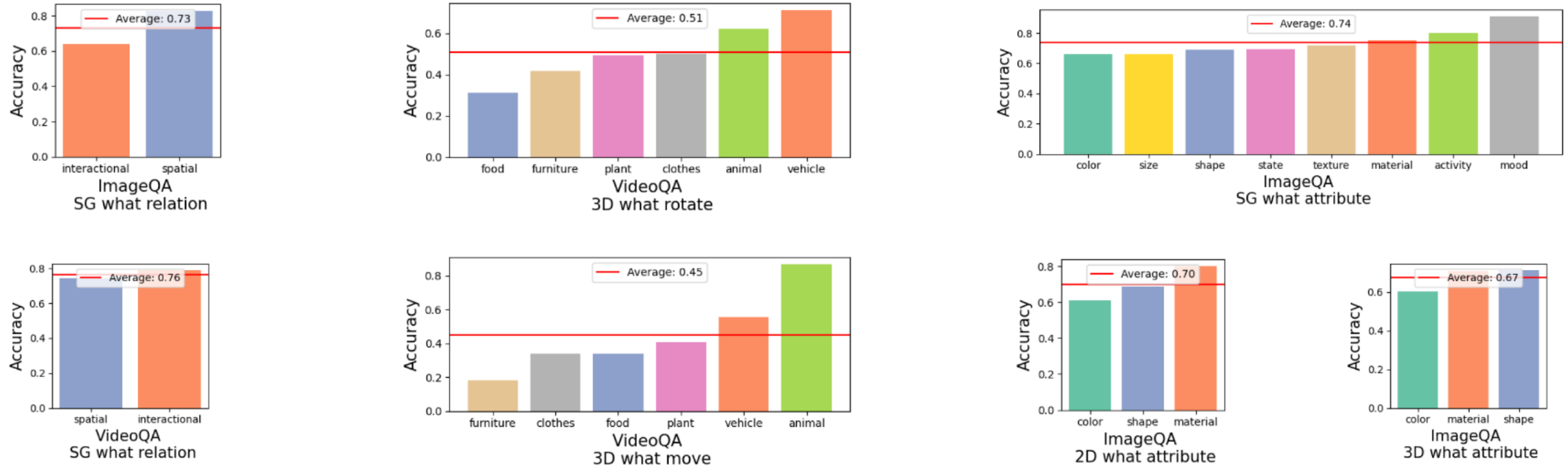
Georgia Tech

Q7: What is today's popular proprietary model (GPT4o) bad at?

(Q1) What relations are GPT4o bad at understanding?

(Q2) What objects are GPT4o bad at recognizing when rotating/moving?

(Q3) What attribute of objects are GPT4o bad at recognizing?



Spatial relations are harder to maintain as objects/camera angle changes

Models "expect" certain temporal behaviors for different objects

Size of color dictionary is significantly larger than mood dictionary

Que	Tas	elations/attributes		Δ Perf. (%)
what whe	Vic Vic	duct, hamper, tool, urniture, air condit	t	-21.67 -19.33
what	Im Vic	g, pushing, pushed by, carrying, abov		-51.05 -16.66
what attributes are GPT4o bad at recognizing?	ImageQA 2D what attribute ImageQA 3D what attribute ImageQA SG what attribute	*purple, brown, red, gray, beige* *stone, rubber, textile, leather, plastic* *crooked, power, lower, steep, glowing*		-5.33 -10.67 -45.45

rgia
h.

Discussion

Limitations and Societal Implications

- **Generated tasks can be unrealistic and biased:** might not capture the nuances of real-world scenarios

- **Designing task space is challenging:** identifying the relevant attributes for each task type might require domain knowledge

- **Adding new task generators requires technical expertise**

- **Inaccuracies in Query results:** Efficient query results approximation within certain budgets might sometimes yield inaccurate results, especially when the budget limits are constrained

Limitations and Societal Implications

- **Misuse for malicious benchmarks:** create adversarial examples to trick or expose vulnerabilities in AI systems.

- **Reinforcing biases and discrimination:** if task generators are not carefully designed, they could perpetuate biases in source data

- **Overreliance on synthetic tasks:** might create a false sense of progress and hinder the development of AI models that effectively address real-world challenges.

- **Data contamination:** Models might learn to exploit patterns in synthetic data and fail to generalize to real-world scenarios.

- **Access and fairness:** Requires technical expertise to create new task generators

Summary of Strengths and Weaknesses

- Strengths:
  - Provide a systematic approach for evaluating different MLMs for user-specific task requirements.
  - Enable users to provide fine-grained queries (e.g., top-k, threshold) for task generation.
  - Provide a database with different open-source and proprietary MLMs evaluated on several benchmarks.
  - Enable users to evaluate different MLMs on fixed computational budget using approaches such as fitting and active learning.
- Weaknesses:
  - Synthetically generated data might not capture the nuances of real-world scenarios.
  - MLMs might learn to exploit specific patterns in synthetic data (especially when trained at scale) and may not generalize well for practical applications.
  - Inaccuracies in estimating model performance for different queries under constrained budget.