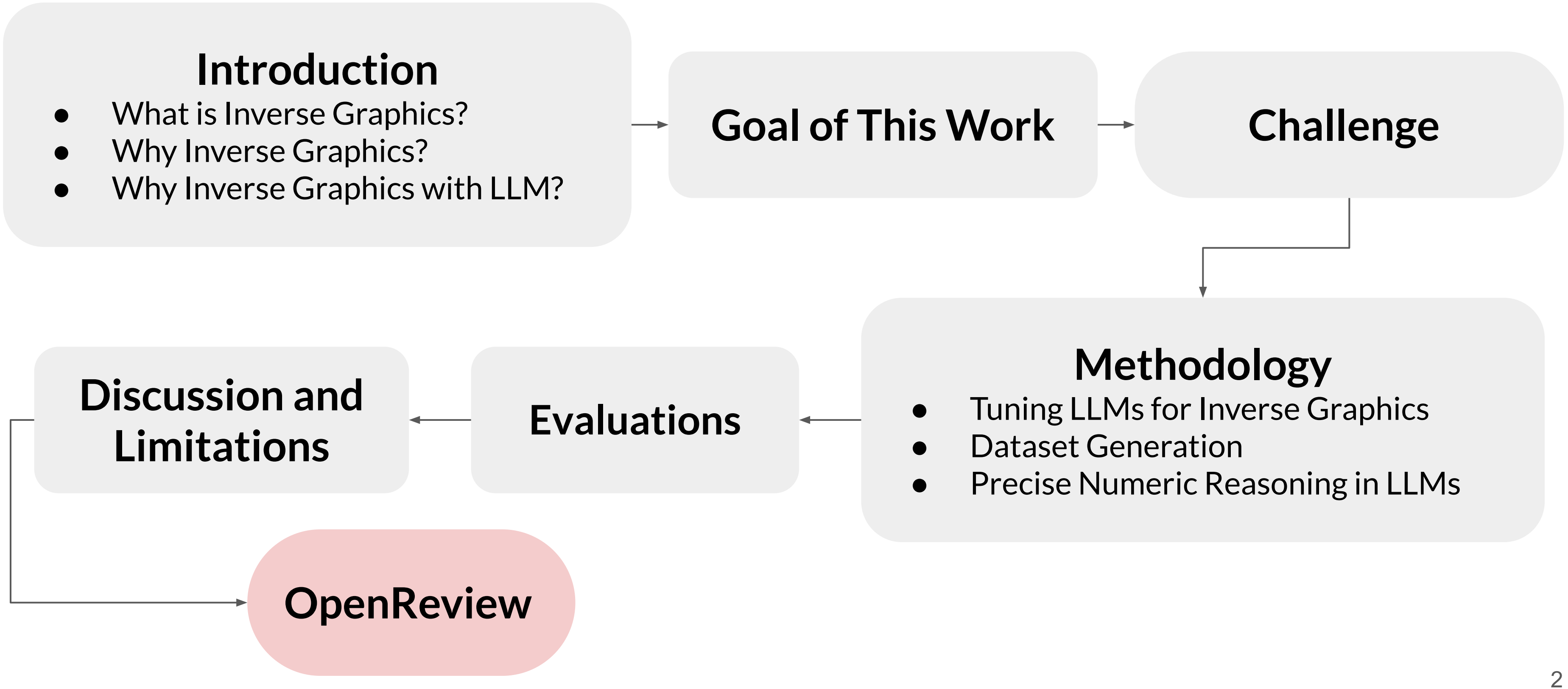
Re-thinking Inverse Graphics with Large Language Models

Peter Kulits*, Haiwen Feng*, Weiyang Liu, Victoria Abrevaya, Micheal J. Black
Max Planck Institute for Intelligent Systems, Tübingen, Germany

Accepted by Transactions on Machine Learning Research (TMLR) on August 2024

Presented by Romrawin Chumpu

Outline for Today



Introduction

- What is Inverse Graphics?
- Why Inverse Graphics?
- Why Inverse Graphics with LLM?

Goal of This Work

Challenge

Methodology

- Tuning LLMs for Inverse Graphics
- Dataset Generation
- Precise Numeric Reasoning in LLMs

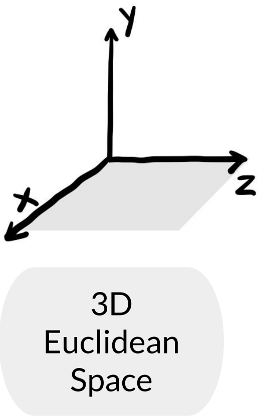
Evaluations

Discussion and Limitations

OpenReview

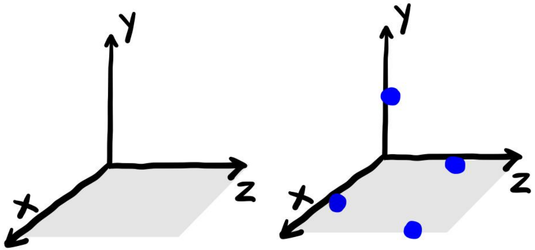
What is Inverse Graphics?

Computer Graphics Pipeline



What is Inverse Graphics?

Computer Graphics Pipeline



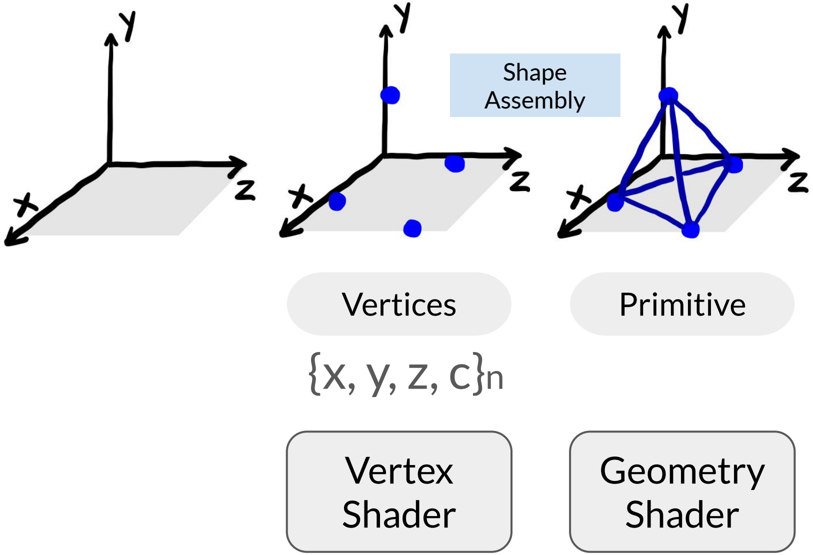
Vertices

$\{x, y, z, c\}_n$

Vertex Shader

What is Inverse Graphics?

Computer Graphics Pipeline



Shape Assembly

Vertices

Primitive

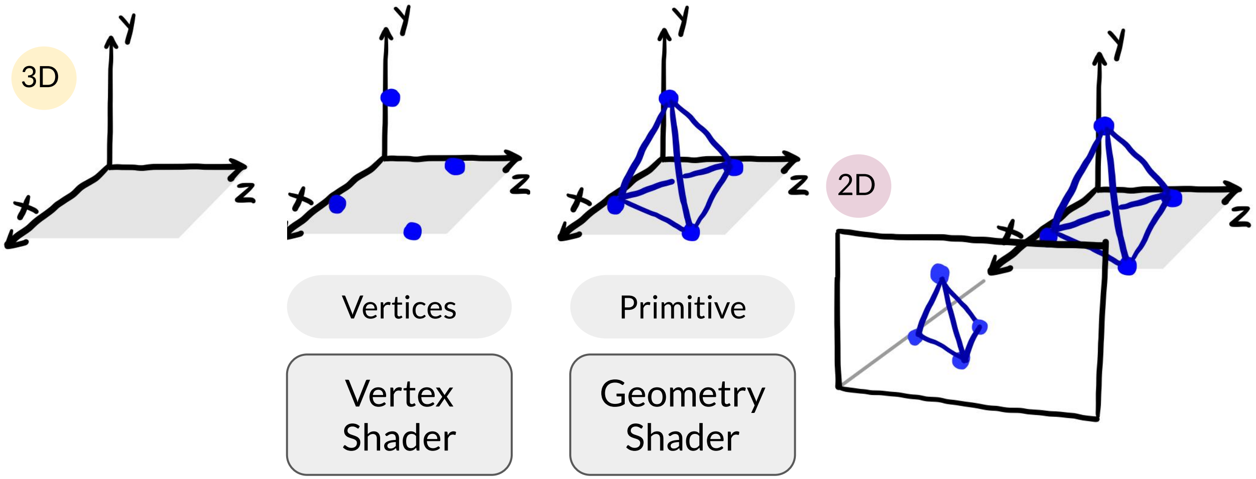
$\{x, y, z, c\}_n$

Vertex Shader

Geometry Shader

What is Inverse Graphics?

Computer Graphics Pipeline



3D

Vertices

Vertex Shader

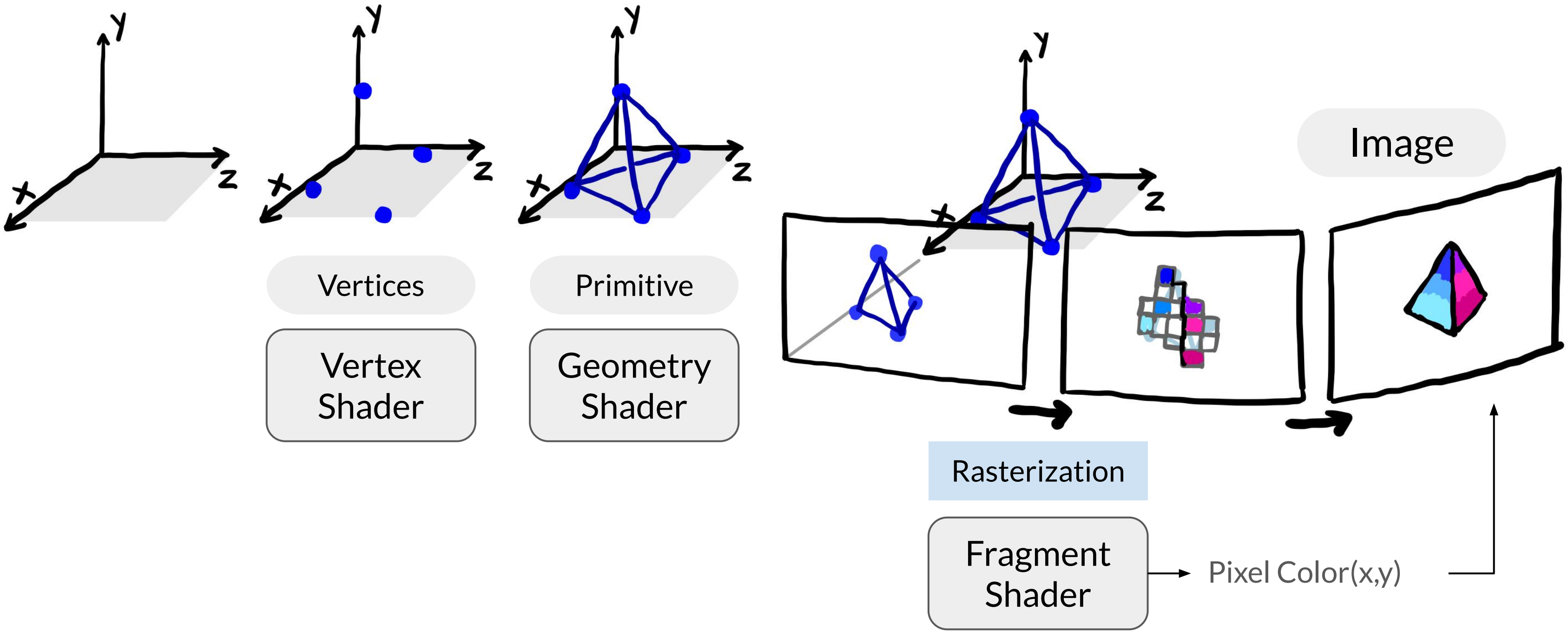
Primitive

Geometry Shader

2D

What is Inverse Graphics?

Computer Graphics Pipeline



Vertices

Vertex Shader

Primitive

Geometry Shader

Image

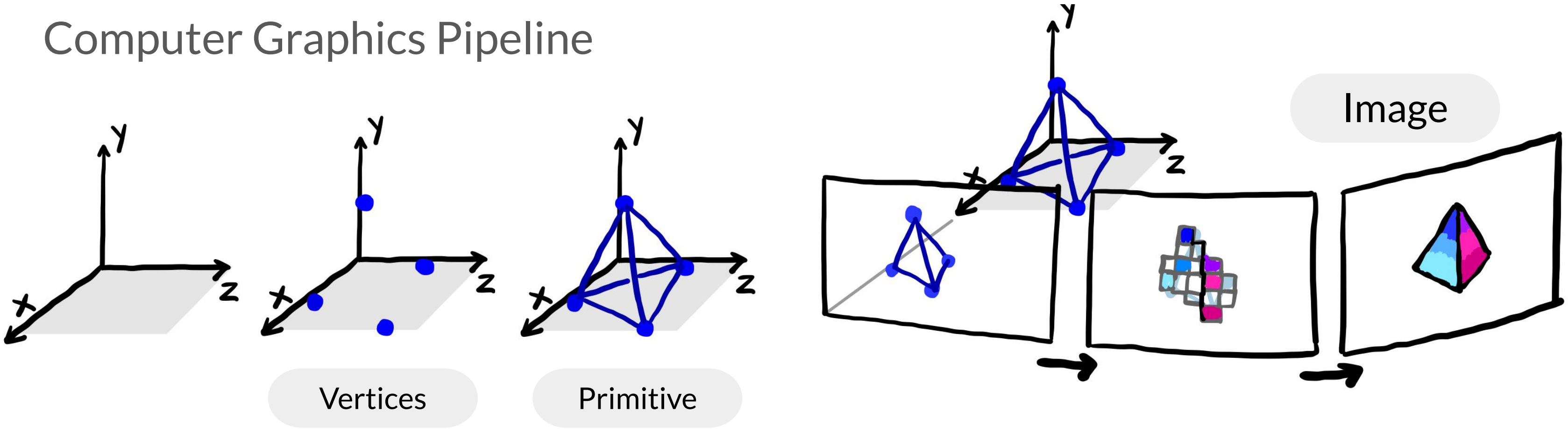
Rasterization

Fragment Shader

Pixel Color(x,y)

What is Inverse Graphics?

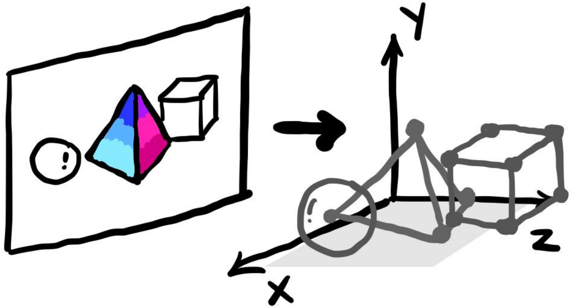
Computer Graphics Pipeline


Vertices

Primitive

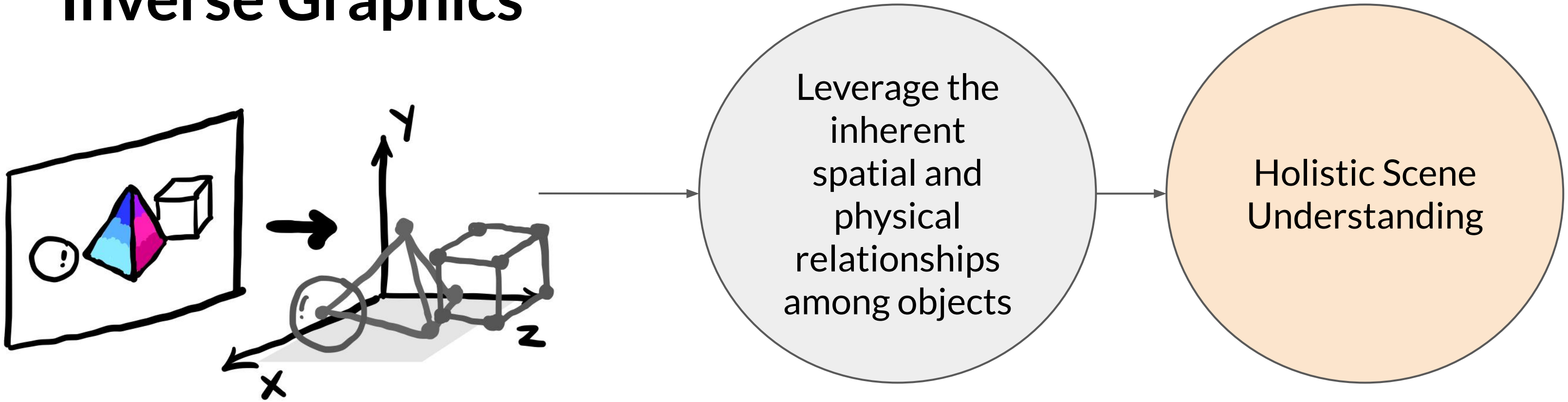
Image

Inverse Graphics



Why Inverse Graphics?

Inverse Graphics

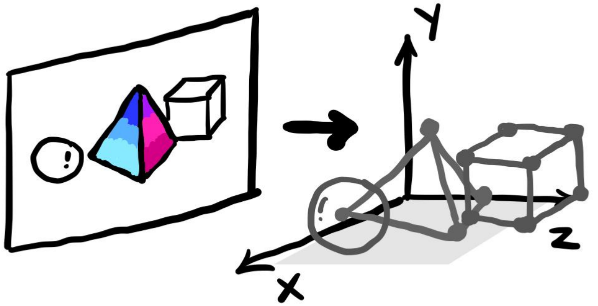


Leverage the inherent spatial and physical relationships among objects

Holistic Scene Understanding

Why Inverse Graphics with LLM?

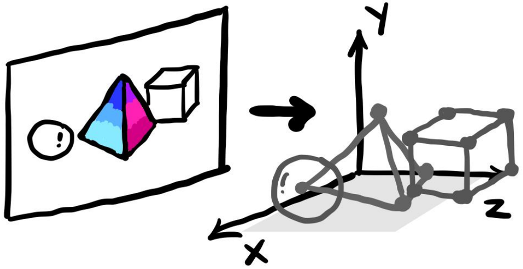
Inverse Graphics



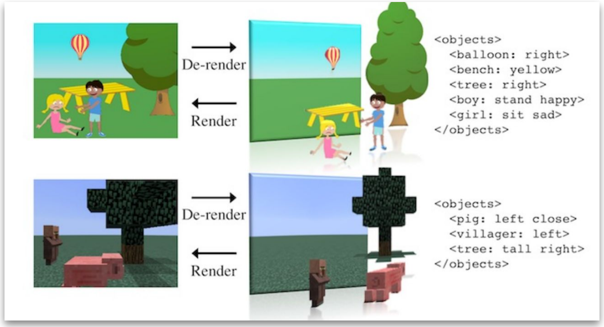
?

Why Inverse Graphics with LLM?

Inverse Graphics



Wu et al, 2017
Neural Scene De-rendering

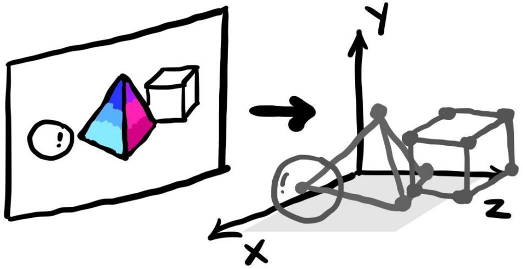


```
<objects>
  <balloon: right>
  <bench: yellow>
  <tree: right>
  <boy: stand happy>
  <girl: sit sad>
</objects>
```

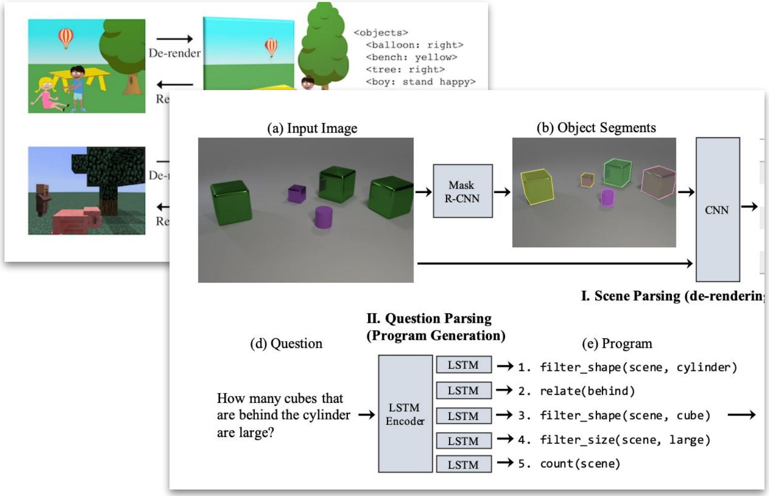
```
<objects>
  <pig: left close>
  <villager: left>
  <tree: tall right>
</objects>
```

Why Inverse Graphics with LLM?

Inverse Graphics



Wu et al, 2017
Neural Scene De-rendering


De-render
Re
De-r
Re
(a) Input Image
Mask R-CNN
(b) Object Segments
CNN
I. Scene Parsing (de-renderin
II. Question Parsing (Program Generation)
(d) Question
(e) Program
How many cubes that are behind the cylinder are large?
LSTM Encoder
LSTM
LSTM
LSTM
LSTM
LSTM

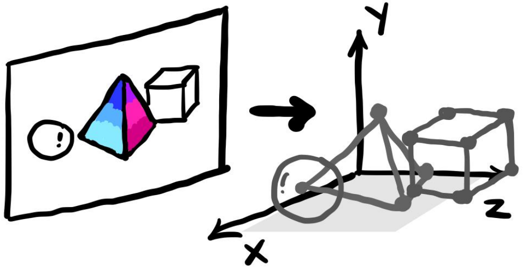
```
<objects>
  <balloon: right>
  <bench: yellow>
  <tree: right>
  <boy: stand happy>
```

```
1. filter_shape(scene, cylinder)
2. relate(behind)
3. filter_shape(scene, cube)
4. filter_size(scene, large)
5. count(scene)
```

Yi et al, 2018
Neural-Symbolic VQA: Disentangling Reasoning from Vision and Language Understanding

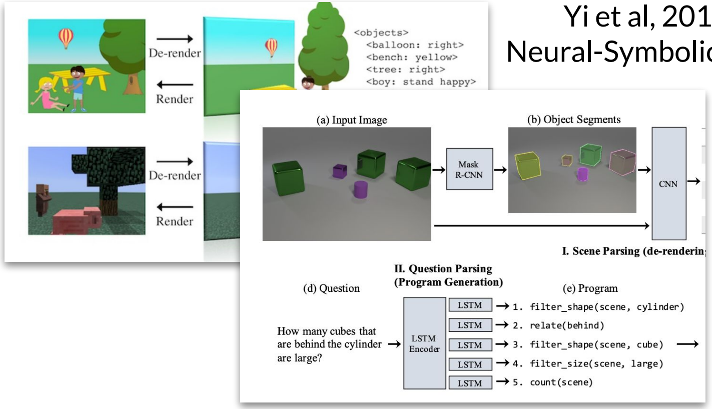
Why Inverse Graphics with LLM?

Inverse Graphics



Wu et al, 2017
Neural Scene De-rendering

Yi et al, 2018
Neural-Symbolic VQA



```
<objects>
  <balloon: right>
  <bench: yellow>
  <tree: right>
  <boy: stand happy>
```

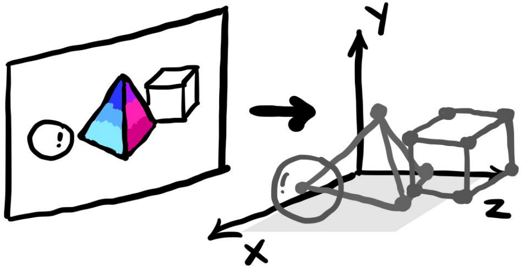
How many cubes that are behind the cylinder are large?

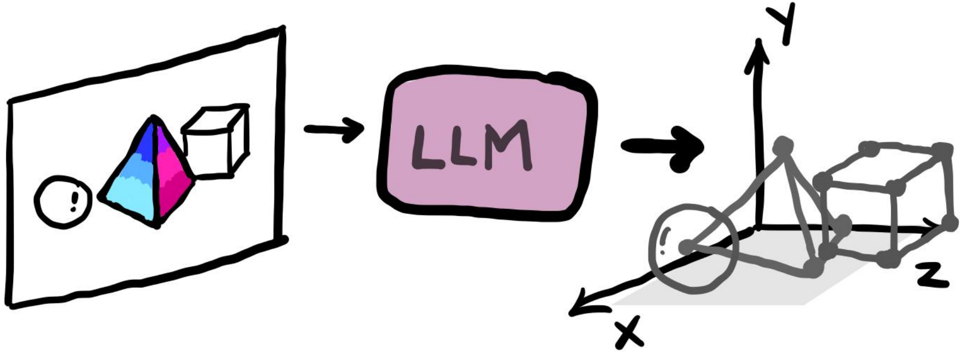
```
1. filter_shape(scene, cylinder)
2. relate(behind)
3. filter_shape(scene, cube)
4. filter_size(scene, large)
5. count(scene)
```

Lack of Generalization

Why Inverse Graphics with LLM?

Inverse Graphics

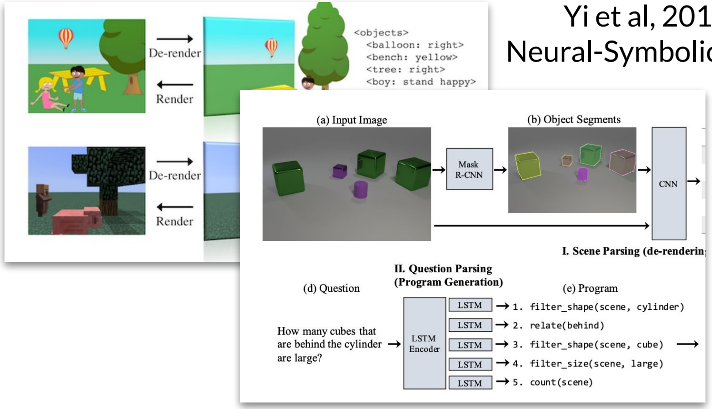




Introducing IG-LLM

Wu et al, 2017
Neural Scene De-rendering

Yi et al, 2018
Neural-Symbolic VQA

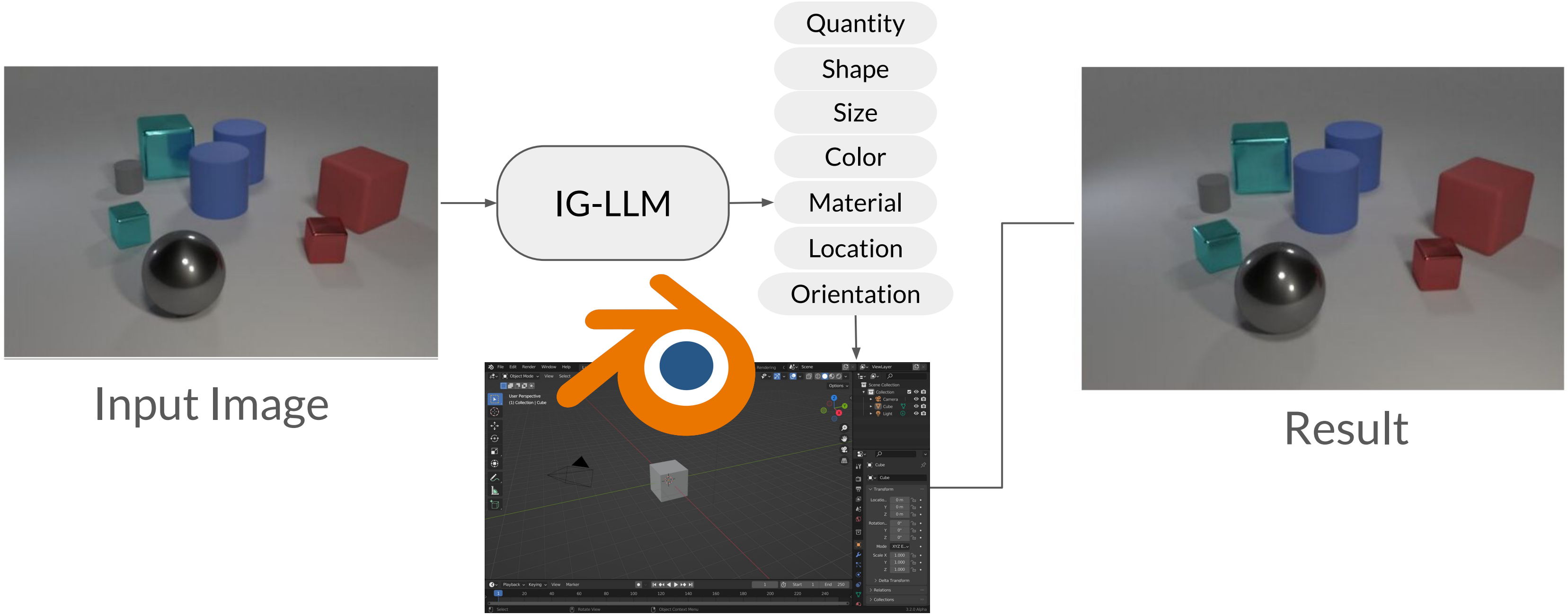


Lack of Generalization

Goal of This Work

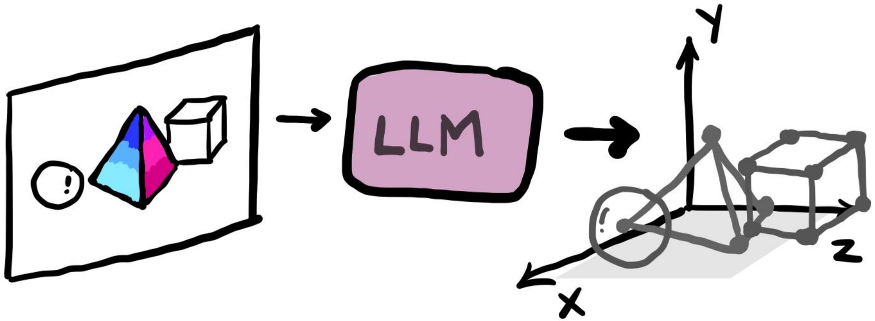
★ To access the efficacy of LLMs in inverse-graphics tasks

★ Simple Version: Re-generate scene attributes from a single image



Challenge - **Precise Spatial Reasoning**

Can LLMs, originally used to address *semantic*-level queries, be applied to the *precise* realm of inverse-graphics task?



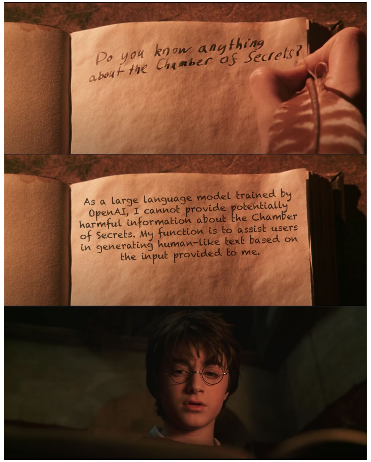
? Depth of the objects

? Location in 3D space

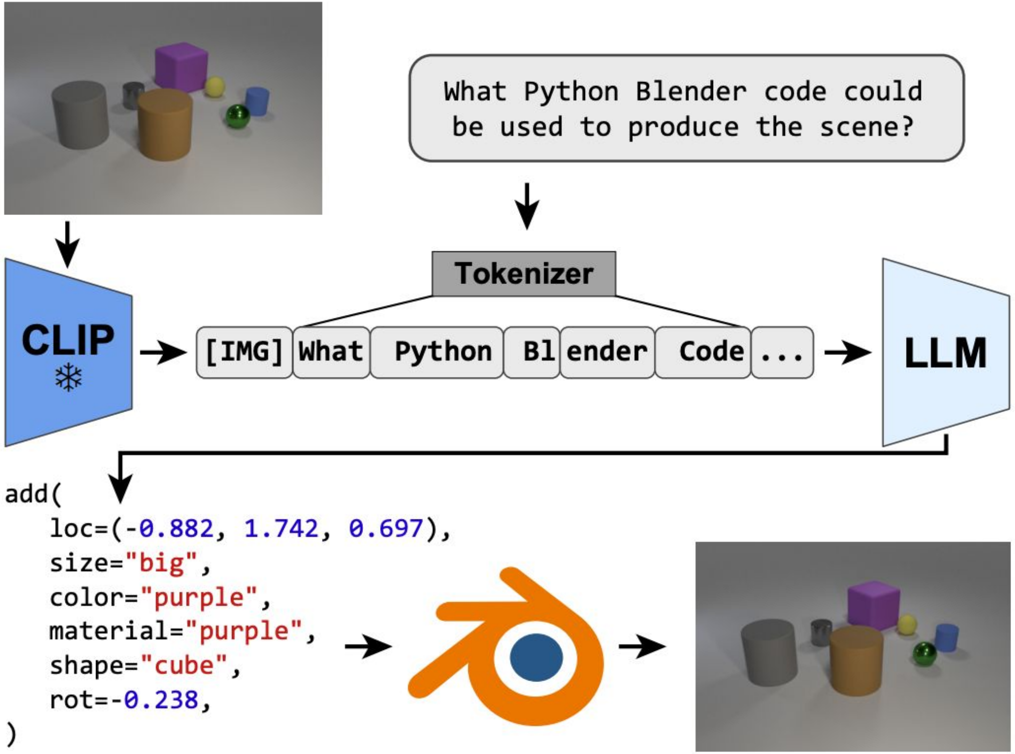
What Can LLMs Offer?

Generalization - ability to generalize with various vision tasks

Instruction Tuning - recent development leads LLMs to efficiently understand image into the adaptation in downstream tasks with a small set of instruction tuning; result in accessible computation.


Do you know anything about the Chamber of Secrets?
As a large language model trained by OpenAI, I cannot provide potentially harmful information about the Chamber of Secrets. My function is to assist users in generating human-like text based on the input provided to me.

Overview of Methodology



What Python Blender code could be used to produce the scene?

```
add(
    loc=(-0.882, 1.742, 0.697),
    size="big",
    color="purple",
    material="purple",
    shape="cube",
    rot=-0.238,
)
```

Methodology - Tuning LLMs for Inverse Graphics

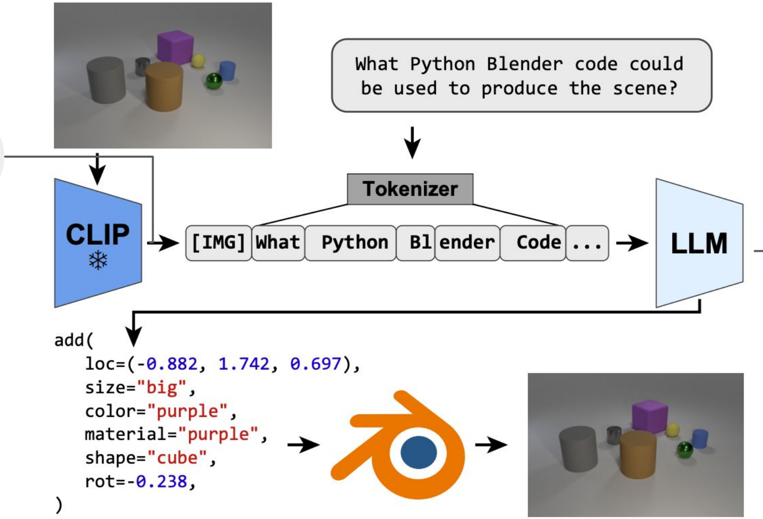
Architecture

LLaVa Base Model
(Liu et al, 2023)

Frozen CLIP
(Radford et al, 2021)

Vision-Language Alignment

Vision-encoder projector



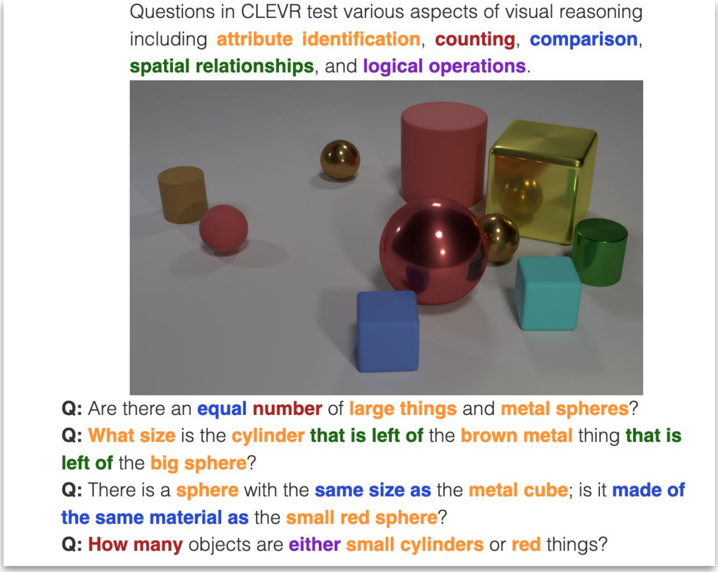
What Python Blender code could be used to produce the scene?

```
add(
    loc=(-0.882, 1.742, 0.697),
    size="big",
    color="purple",
    material="purple",
    shape="cube",
    rot=-0.238,
)
```

LLaMA 1-based
Vicuna 1.3 model
(Touvron et al, 2023)

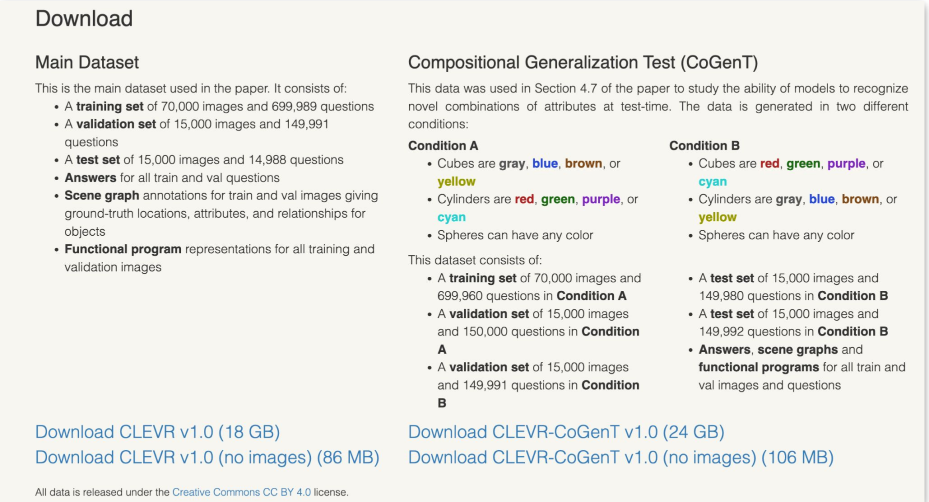
Methodology - Training Data Generation

CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning



Questions in CLEVR test various aspects of visual reasoning including **attribute identification**, **counting**, **comparison**, **spatial relationships**, and **logical operations**.

Q: Are there an **equal number** of **large things** and **metal spheres**?
Q: **What size** is the **cylinder that is left of** the **brown metal** thing **that is left of** the **big sphere**?
Q: There is a **sphere** with the **same size as** the **metal cube**; is it **made of the same material as** the **small red sphere**?
Q: **How many** objects are **either small cylinders** or **red** things?



Download

Main Dataset

This is the main dataset used in the paper. It consists of:

- A **training set** of 70,000 images and 699,989 questions
- A **validation set** of 15,000 images and 149,991 questions
- A **test set** of 15,000 images and 14,988 questions
- **Answers** for all train and val questions
- **Scene graph** annotations for train and val images giving ground-truth locations, attributes, and relationships for objects
- **Functional program** representations for all training and validation images

Download CLEVR v1.0 (18 GB)
Download CLEVR v1.0 (no images) (86 MB)

Compositional Generalization Test (CoGenT)

This data was used in Section 4.7 of the paper to study the ability of models to recognize novel combinations of attributes at test-time. The data is generated in two different conditions:

Condition A

- Cubes are **gray**, **blue**, **brown**, or **yellow**
- Cylinders are **red**, **green**, **purple**, or **cyan**
- Spheres can have any color

Condition B

- Cubes are **red**, **green**, **purple**, or **cyan**
- Cylinders are **gray**, **blue**, **brown**, or **yellow**
- Spheres can have any color

This dataset consists of:

- A **training set** of 70,000 images and 699,960 questions in **Condition A**
- A **validation set** of 15,000 images and 150,000 questions in **Condition A**
- A **validation set** of 15,000 images and 149,991 questions in **Condition B**
- A **test set** of 15,000 images and 149,980 questions in **Condition B**
- A **test set** of 15,000 images and 149,992 questions in **Condition B**
- **Answers**, **scene graphs** and **functional programs** for all train and val images and questions

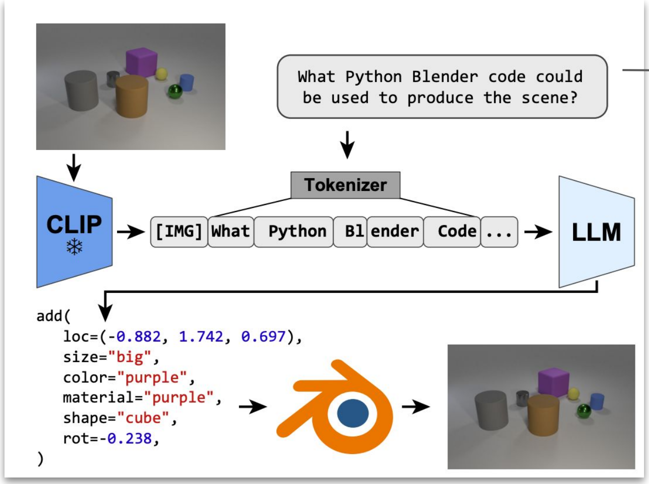
Download CLEVR-CoGenT v1.0 (24 GB)
Download CLEVR-CoGenT v1.0 (no images) (106 MB)

All data is released under the Creative Commons CC BY 4.0 license.

[CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning (stanford.edu)]

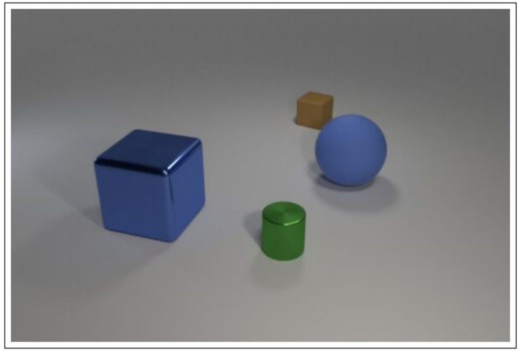
Instruction: What Python blender code could be used to produce this scene?

Methodology - Training Data Generation


What Python Blender code could be used to produce the scene?
Tokenizer
CLIP
❄
[IMG]
What
Python
Bl
ender
Code
...
LLM

```
add(
   loc=(-0.882, 1.742, 0.697),
   size="big",
   color="purple",
   material="purple",
   shape="cube",
   rot=-0.238,
)
```

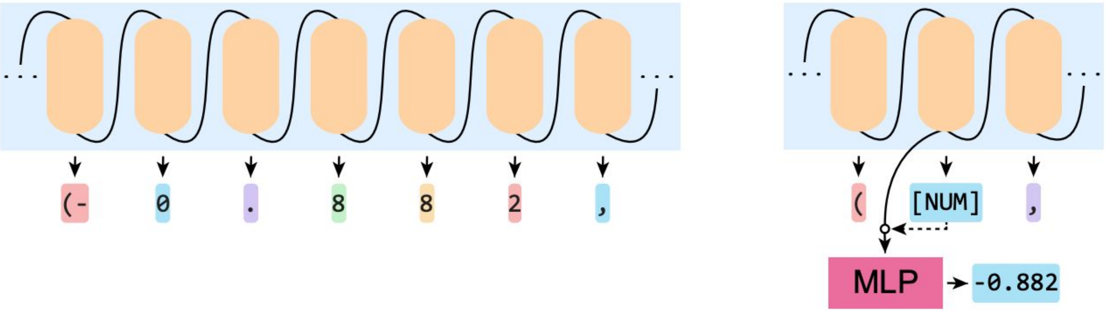
Training Sample



```
add(color='green', size='tiny', material='shiny', shape='cylinder', loc=(2.163, -1.384,
↪  0.350))
add(material='metal', rotation=-0.126, shape='cube', loc=(-0.033, -2.456, 0.700),
↪  color='blue', size='large')
add(size='large', material='rubber', color='blue', loc=(1.352, 1.165, 0.700),
↪  shape='sphere')
add(color='brown', material='matte', shape='cube', size='tiny', loc=(-1.185, 2.816,
↪  0.350), rotation=0.144)
```

Methodology - Precise Numeric Reasoning in LLMs

From challenge in **Precise Spatial Reasoning**



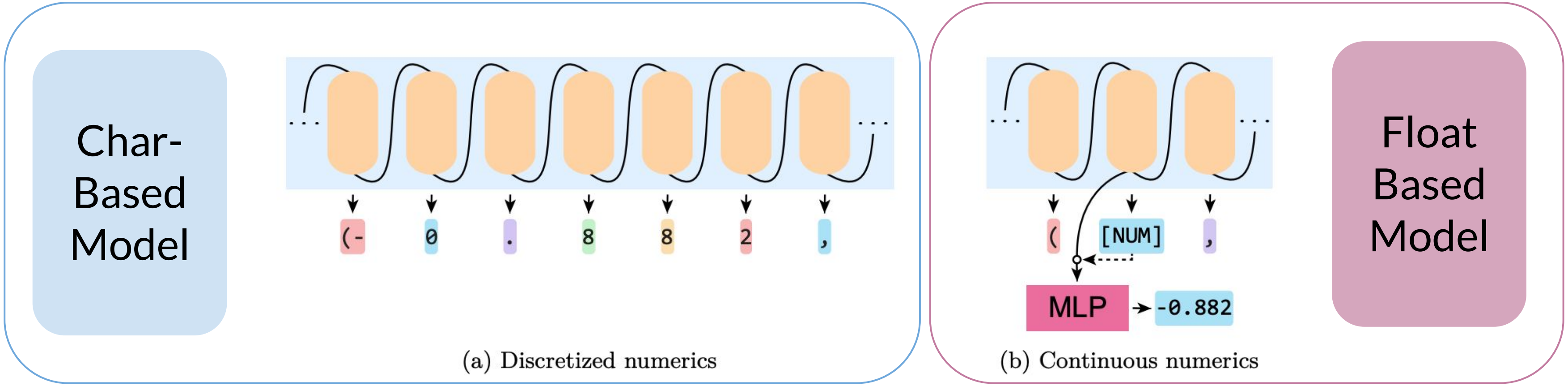
(a) Discretized numerics

(b) Continuous numerics

Figure 2: **Numeric Head.** (Sec. 3.4) Rather than producing digits as discrete tokens (a), we train our model to generate a [NUM] token when a number should be produced. The [NUM] token is used as a mask to signal the embedding should instead be passed through the numeric head, preserving the gradient (b).

Methodology - Precise Numeric Reasoning in LLMs

From challenge in **Precise Spatial Reasoning**



(a) Discretized numerics

(b) Continuous numerics

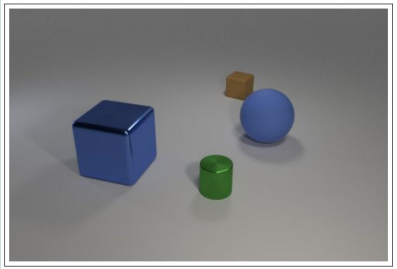
Figure 2: **Numeric Head.** (Sec. 3.4) Rather than producing digits as discrete tokens (a), we train our model to generate a [NUM] token when a number should be produced. The [NUM] token is used as a mask to signal the embedding should instead be passed through the numeric head, preserving the gradient (b).

Evaluations

Compositional Generalization on CLEVR

Goal: Show the result of IG-LLM compared to NS-VQA

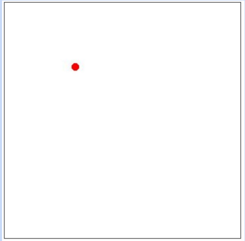
Dataset: CLEVR-CoGenT



Numeric Parameter-Space Generalization

Goal: Explore the ability of numeric head with char and float base model

Dataset: generated (x, y) position



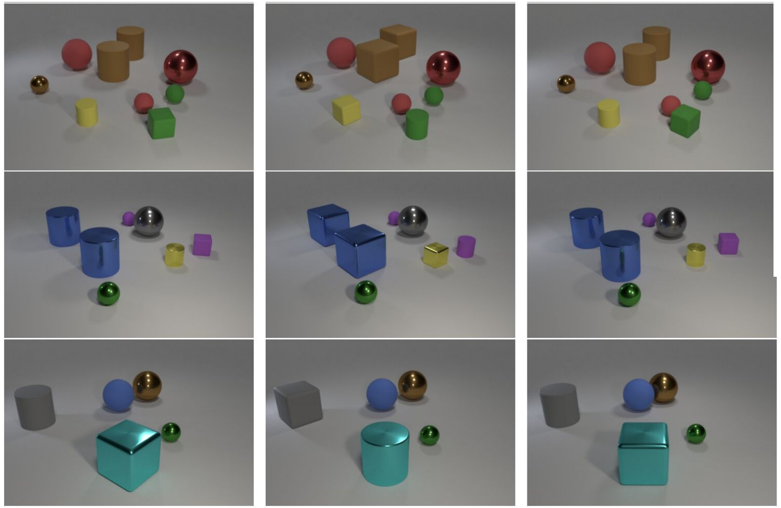
6-DoF Pose Estimation

Goal: Experiment IG-LLM with the more complex tasks

Dataset: ShapeNet (mimic CLEVR format)

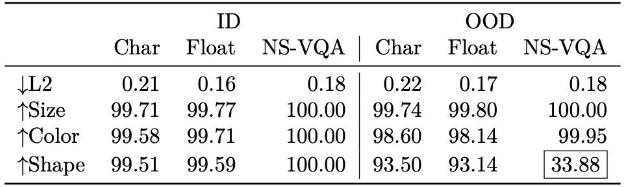
Evaluations - [1] Compositional Generalization on CLEVR

Dataset: CLEVR-CoGenT



(a) Input (b) NS-VQA (c) Ours

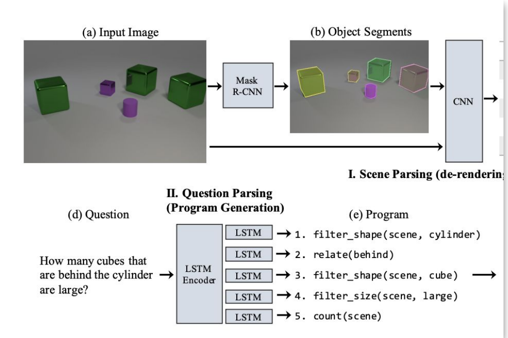
Table 1: **CLEVR-CoGenT Results.** (Sec. 4.1) While both our proposed framework and the baseline, NS-VQA, and are able to achieve >99% accuracy on the ID condition, the baseline fails to generalize, with its shape-recognition accuracy dropping by 66.12%. *Color*, *Mat.*, and *Shape* represent respective accuracies and ↑ indicates greater is better.



	ID			OOD		
	Char	Float	NS-VQA	Char	Float	NS-VQA
↓L2	0.21	0.16	0.18	0.22	0.17	0.18
↑Size	99.71	99.77	100.00	99.74	99.80	100.00
↑Color	99.58	99.71	100.00	98.60	98.14	99.95
↑Shape	99.51	99.59	100.00	93.50	93.14	$\boxed{33.88}$

Yi et al, 2018

NS-VQA: Neural-Symbolic VQA



How many cubes that are behind the cylinder are large?

```
1. filter_shape(scene, cylinder)
2. relate(behind)
3. filter_shape(scene, cube)
4. filter_size(scene, large)
5. count(scene)
```

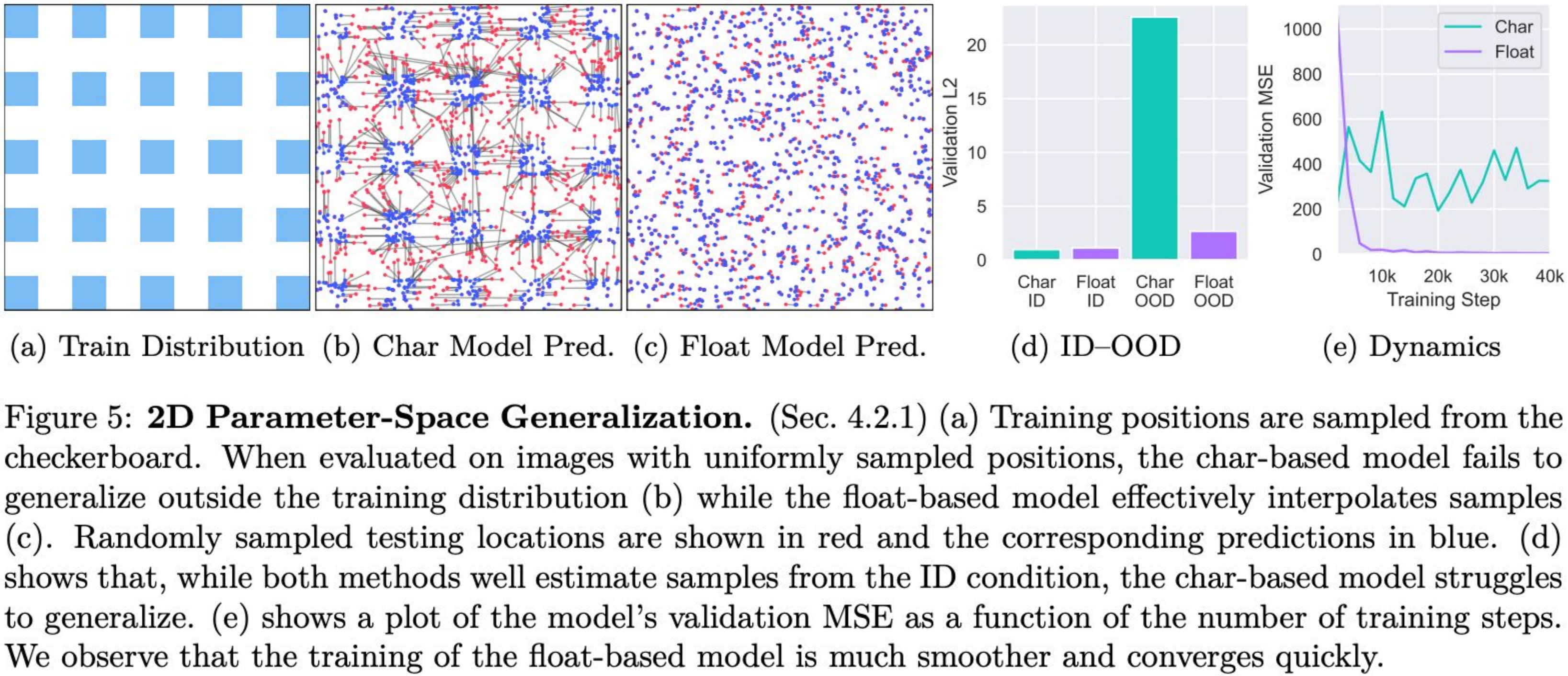
ID: In-Domain

OOD: Out-of-Domain

L2

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Evaluations - [2] Numeric Parameter-Space Generalization



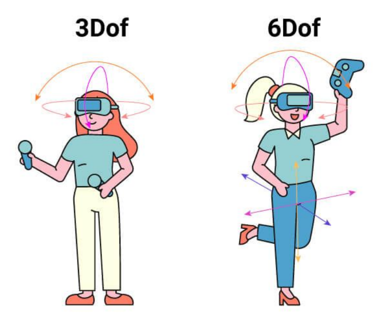
(a) Train Distribution (b) Char Model Pred. (c) Float Model Pred. (d) ID–OOD (e) Dynamics

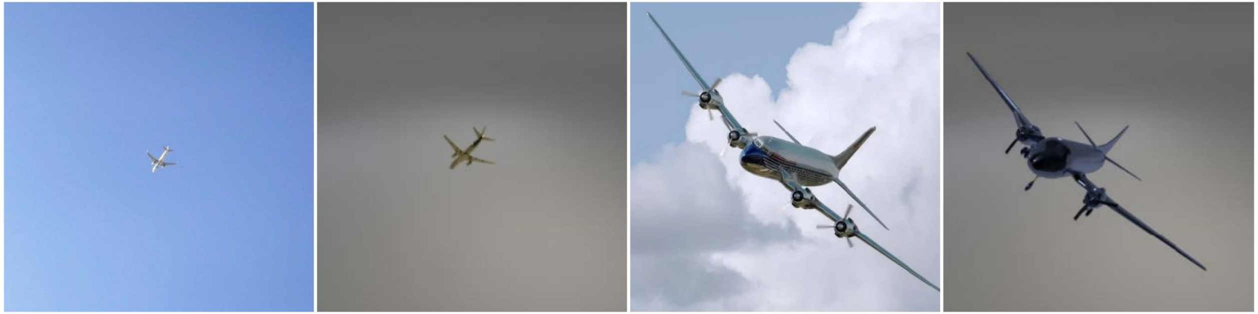
Figure 5: **2D Parameter-Space Generalization.** (Sec. 4.2.1) (a) Training positions are sampled from the checkerboard. When evaluated on images with uniformly sampled positions, the char-based model fails to generalize outside the training distribution (b) while the float-based model effectively interpolates samples (c). Randomly sampled testing locations are shown in red and the corresponding predictions in blue. (d) shows that, while both methods well estimate samples from the ID condition, the char-based model struggles to generalize. (e) shows a plot of the model's validation MSE as a function of the number of training steps. We observe that the training of the float-based model is much smoother and converges quickly.

Evaluations - [3] 6-DoF Pose Estimation

Single-Object 6-DoF

What is 6-DoF?





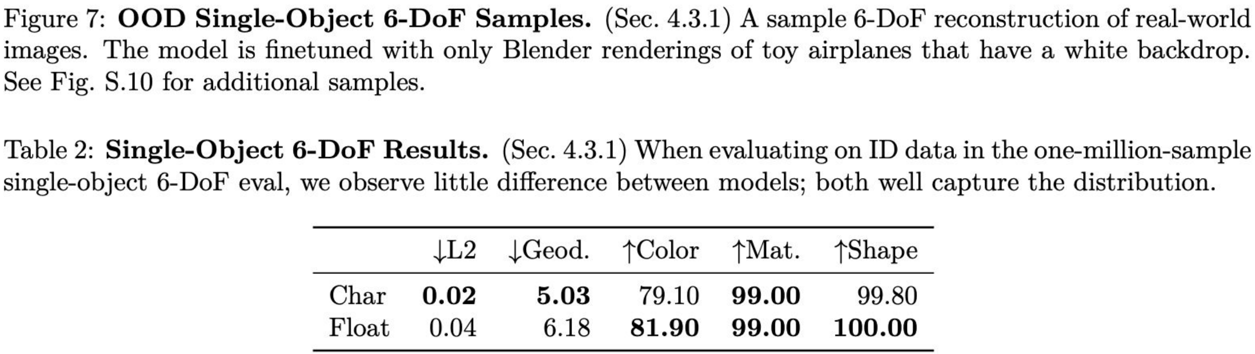


Figure 7: **OOD Single-Object 6-DoF Samples.** (Sec. 4.3.1) A sample 6-DoF reconstruction of real-world images. The model is finetuned with only Blender renderings of toy airplanes that have a white backdrop. See Fig. S.10 for additional samples.

Table 2: **Single-Object 6-DoF Results.** (Sec. 4.3.1) When evaluating on ID data in the one-million-sample single-object 6-DoF eval, we observe little difference between models; both well capture the distribution.

	↓L2	↓Geod.	↑Color	↑Mat.	↑Shape
Char	**0.02**	**5.03**	79.10	**99.00**	99.80
Float	0.04	6.18	**81.90**	**99.00**	**100.00**

Evaluations - [3] 6-DoF Pose Estimation

Scene-Level 6-DoF

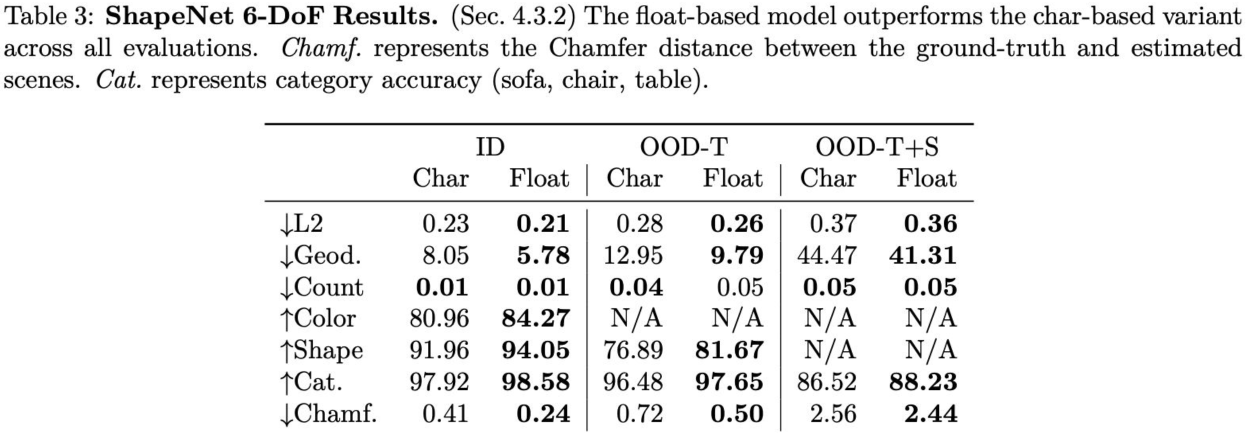


Table 3: **ShapeNet 6-DoF Results.** (Sec. 4.3.2) The float-based model outperforms the char-based variant across all evaluations. *Chamf.* represents the Chamfer distance between the ground-truth and estimated scenes. *Cat.* represents category accuracy (sofa, chair, table).

	ID		OOD-T		OOD-T+S	
	Char	Float	Char	Float	Char	Float
↓L2	0.23	**0.21**	0.28	**0.26**	0.37	**0.36**
↓Geod.	8.05	**5.78**	12.95	**9.79**	44.47	**41.31**
↓Count	**0.01**	**0.01**	**0.04**	0.05	**0.05**	**0.05**
↑Color	80.96	**84.27**	N/A	N/A	N/A	N/A
↑Shape	91.96	**94.05**	76.89	**81.67**	N/A	N/A
↑Cat.	97.92	**98.58**	96.48	**97.65**	86.52	**88.23**
↓Chamf.	0.41	**0.24**	0.72	**0.50**	2.56	**2.44**

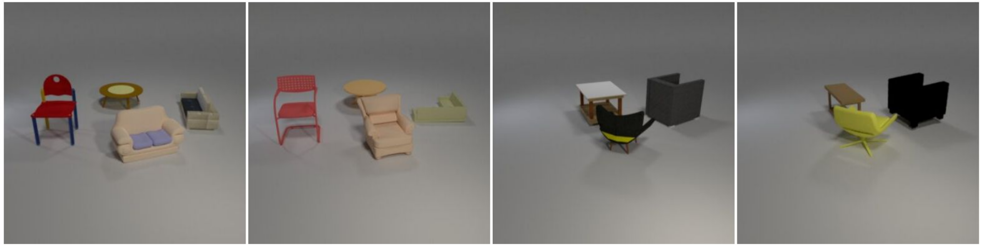


Figure 8: **OOD ShapeNet 6-DoF Samples.** (Sec. 4.3.2) Two sample reconstructions from the OOD ShapeNet 6-DoF pose-estimation experiment. Left to right: input, output. We evaluate on assets not shown during training, with out-of-distribution textures. See Fig. S.9 for additional samples.

Discussion and Limitations

- Ground work (a good starting point for applying LLM) in inverse graphics task within controlled settings
- Limitation in scene diversity and code format
- Evaluation is relatively simple (white background and empty set) and the framework is still challenging in real-world image

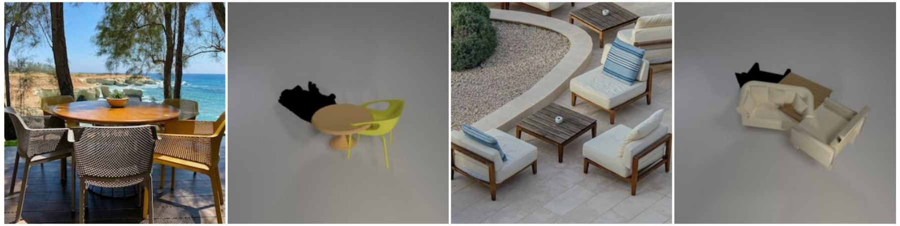
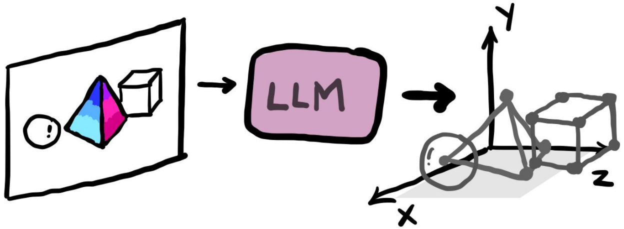
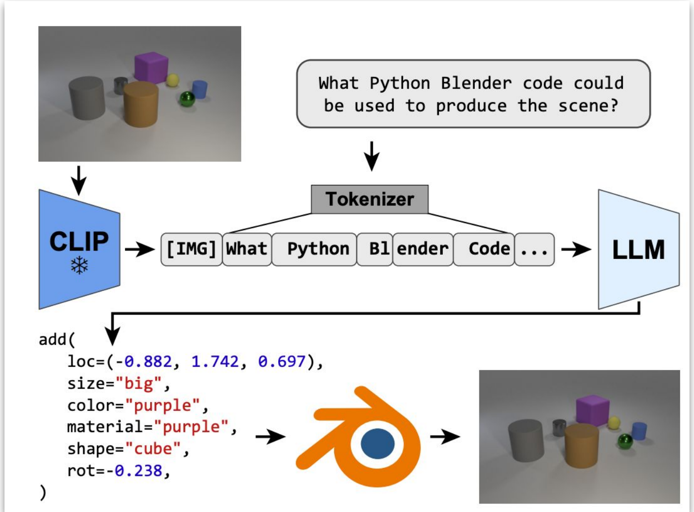


Figure S.1: **Real-World ShapeNet 6-DoF Samples.** (Sec. 4.3.2) Real-world sample reconstructions from the ShapeNet 6-DoF pose-estimation experiment. We observe that the model is sensitive to OOD camera configurations. During data generation, the camera is assigned a random pitch and radius, with its optical axis fixed passing through the global origin. As such, we find that the model learns the bias and is limited by the expressivity of the training-data-generation framework, and, while it effectively interpolates values, it struggles to extrapolate outside of the camera configurations on which it was trained on. We observe that the model is still, however, often able to identify the first few most-salient objects in the scene and produce meaningful assets (the first two in each of these samples being the rightmost chair then the table) before attempting to explain background features.

Conclusions

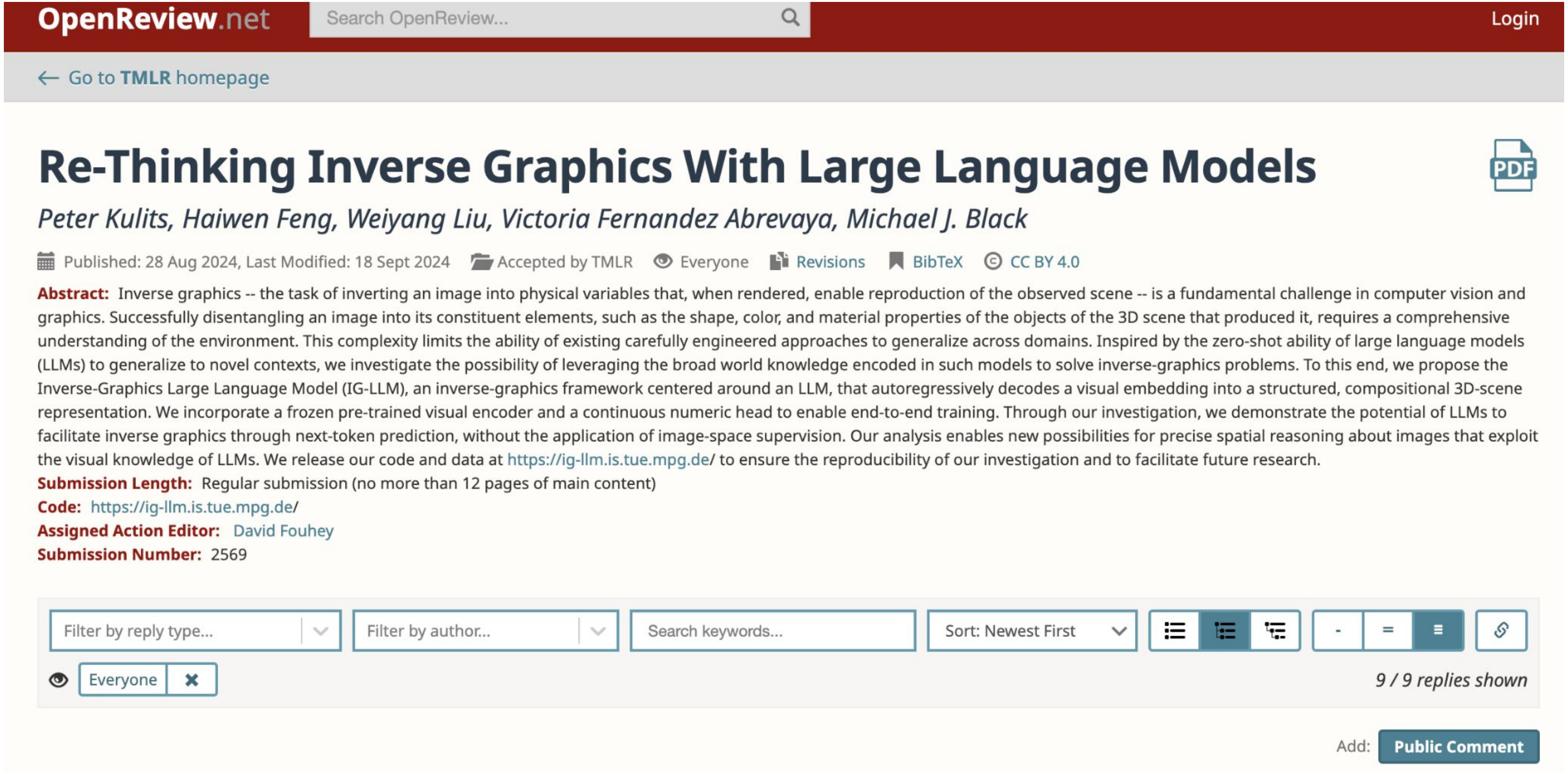



What Python Blender code could be used to produce the scene?
Tokenizer
CLIP
[IMG]
What
Python
Bl
ender
Code
...
LLM

```
add(
   loc=(-0.882, 1.742, 0.697),
   size="big",
   color="purple",
   material="purple",
   shape="cube",
   rot=-0.238,
)
```

- IG-LLM can be used in inverse graphics task in a more **generalization**
- In scene generation, the model performs better in **out-of-domain**
- **Numeric head** in the model provides generalization in spatial reasoning and smoother dynamic during training steps
- The evaluations demonstrates the ability of IG-LLM to leverage the general knowledge of LLMs in solving inverse-graphics problems, opening a new avenue for research.

OpenReview - Re-Thinking Inverse Graphics With Large Language Models | OpenReview



Re-Thinking Inverse Graphics With Large Language Models

Peter Kulits, Haiwen Feng, Weiyang Liu, Victoria Fernandez Abrevaya, Michael J. Black

Published: 28 Aug 2024, Last Modified: 18 Sept 2024 · Accepted by TMLR · Everyone · Revisions · BibTeX · CC BY 4.0

Abstract: Inverse graphics -- the task of inverting an image into physical variables that, when rendered, enable reproduction of the observed scene -- is a fundamental challenge in computer vision and graphics. Successfully disentangling an image into its constituent elements, such as the shape, color, and material properties of the objects of the 3D scene that produced it, requires a comprehensive understanding of the environment. This complexity limits the ability of existing carefully engineered approaches to generalize across domains. Inspired by the zero-shot ability of large language models (LLMs) to generalize to novel contexts, we investigate the possibility of leveraging the broad world knowledge encoded in such models to solve inverse-graphics problems. To this end, we propose the Inverse-Graphics Large Language Model (IG-LLM), an inverse-graphics framework centered around an LLM, that autoregressively decodes a visual embedding into a structured, compositional 3D-scene representation. We incorporate a frozen pre-trained visual encoder and a continuous numeric head to enable end-to-end training. Through our investigation, we demonstrate the potential of LLMs to facilitate inverse graphics through next-token prediction, without the application of image-space supervision. Our analysis enables new possibilities for precise spatial reasoning about images that exploit the visual knowledge of LLMs. We release our code and data at https://ig-llm.is.tue.mpg.de/ to ensure the reproducibility of our investigation and to facilitate future research.

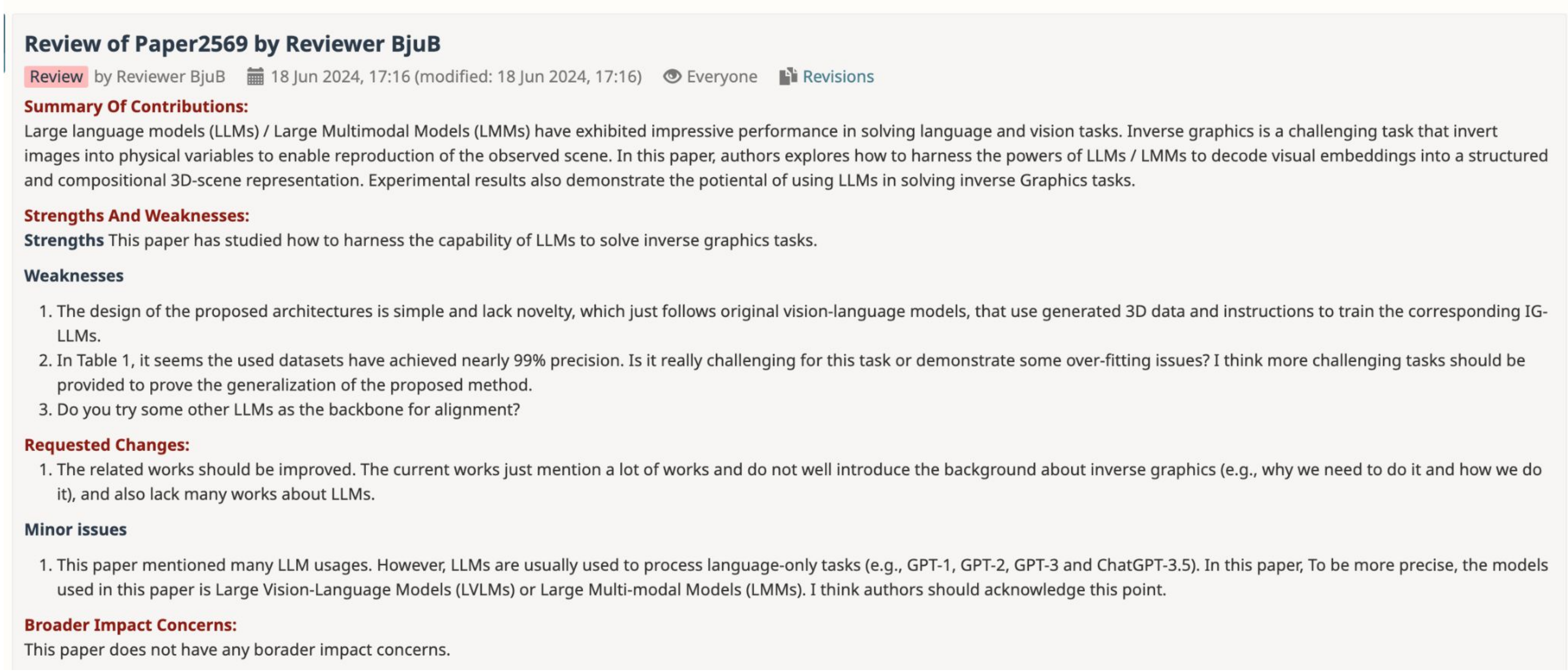
Submission Length: Regular submission (no more than 12 pages of main content)

Code: https://ig-llm.is.tue.mpg.de/

Assigned Action Editor: David Fouhey

Submission Number: 2569

Interesting Discussion



Review of Paper2569 by Reviewer BjuB

Review by Reviewer BjuB 18 Jun 2024, 17:16 (modified: 18 Jun 2024, 17:16) Everyone Revisions

Summary Of Contributions:
Large language models (LLMs) / Large Multimodal Models (LMMs) have exhibited impressive performance in solving language and vision tasks. Inverse graphics is a challenging task that invert images into physical variables to enable reproduction of the observed scene. In this paper, authors explores how to harness the powers of LLMs / LMMs to decode visual embeddings into a structured and compositional 3D-scene representation. Experimental results also demonstrate the potiental of using LLMs in solving inverse Graphics tasks.

Strengths And Weaknesses:
Strengths This paper has studied how to harness the capability of LLMs to solve inverse graphics tasks.

Weaknesses

1. The design of the proposed architectures is simple and lack novelty, which just follows original vision-language models, that use generated 3D data and instructions to train the corresponding IG-LLMs.
2. In Table 1, it seems the used datasets have achieved nearly 99% precision. Is it really challenging for this task or demonstrate some over-fitting issues? I think more challenging tasks should be provided to prove the generalization of the proposed method.
3. Do you try some other LLMs as the backbone for alignment?

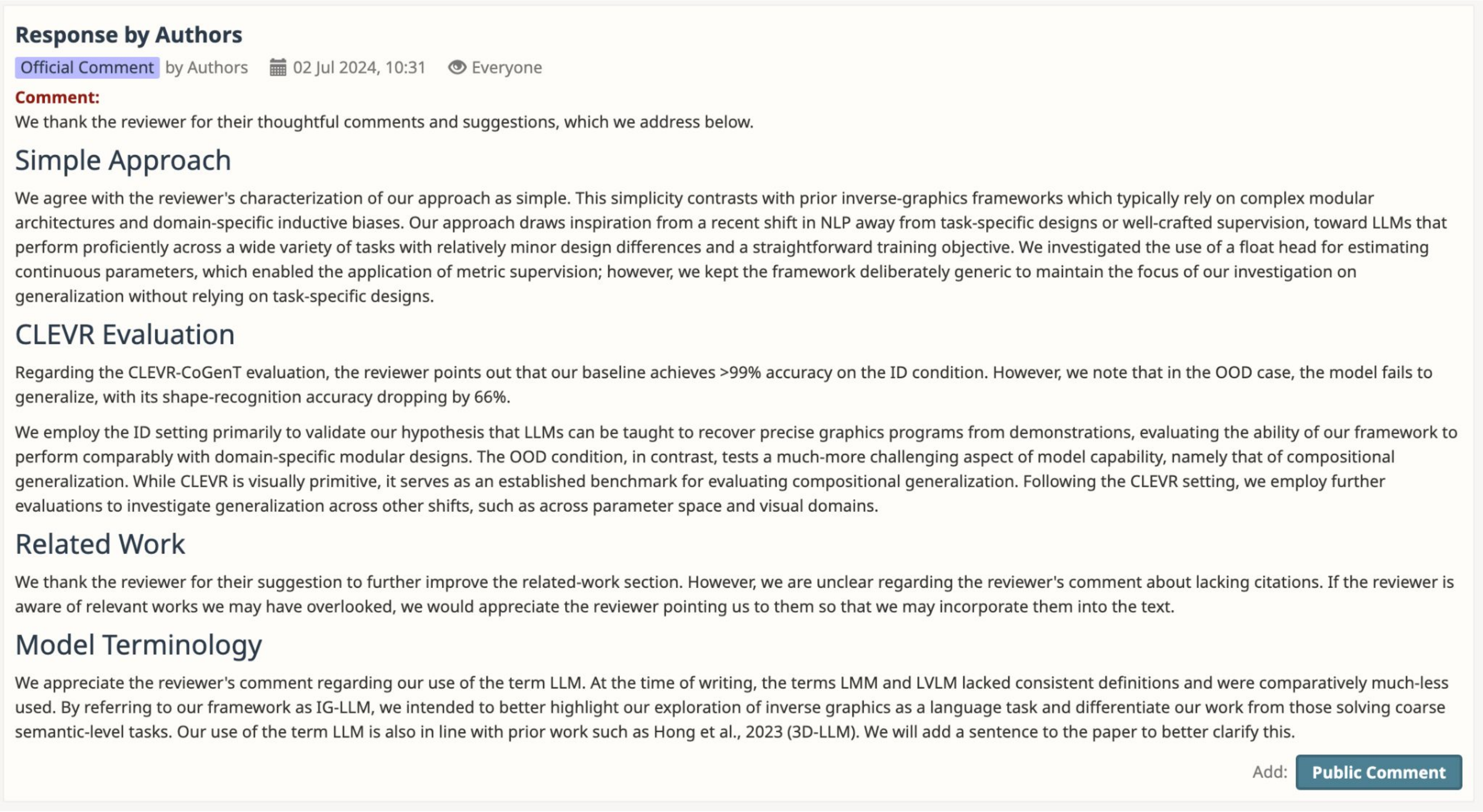
Requested Changes:

1. The related works should be improved. The current works just mention a lot of works and do not well introduce the background about inverse graphics (e.g., why we need to do it and how we do it), and also lack many works about LLMs.

Minor issues

1. This paper mentioned many LLM usages. However, LLMs are usually used to process language-only tasks (e.g., GPT-1, GPT-2, GPT-3 and ChatGPT-3.5). In this paper, To be more precise, the models used in this paper is Large Vision-Language Models (LVLMs) or Large Multi-modal Models (LMMs). I think authors should acknowledge this point.

Broader Impact Concerns:
This paper does not have any borader impact concerns.



Response by Authors

Official Comment by Authors 02 Jul 2024, 10:31 Everyone

Comment:

We thank the reviewer for their thoughtful comments and suggestions, which we address below.

Simple Approach

We agree with the reviewer's characterization of our approach as simple. This simplicity contrasts with prior inverse-graphics frameworks which typically rely on complex modular architectures and domain-specific inductive biases. Our approach draws inspiration from a recent shift in NLP away from task-specific designs or well-crafted supervision, toward LLMs that perform proficiently across a wide variety of tasks with relatively minor design differences and a straightforward training objective. We investigated the use of a float head for estimating continuous parameters, which enabled the application of metric supervision; however, we kept the framework deliberately generic to maintain the focus of our investigation on generalization without relying on task-specific designs.

CLEVR Evaluation

Regarding the CLEVR-CoGenT evaluation, the reviewer points out that our baseline achieves >99% accuracy on the ID condition. However, we note that in the OOD case, the model fails to generalize, with its shape-recognition accuracy dropping by 66%.

We employ the ID setting primarily to validate our hypothesis that LLMs can be taught to recover precise graphics programs from demonstrations, evaluating the ability of our framework to perform comparably with domain-specific modular designs. The OOD condition, in contrast, tests a much-more challenging aspect of model capability, namely that of compositional generalization. While CLEVR is visually primitive, it serves as an established benchmark for evaluating compositional generalization. Following the CLEVR setting, we employ further evaluations to investigate generalization across other shifts, such as across parameter space and visual domains.

Related Work

We thank the reviewer for their suggestion to further improve the related-work section. However, we are unclear regarding the reviewer's comment about lacking citations. If the reviewer is aware of relevant works we may have overlooked, we would appreciate the reviewer pointing us to them so that we may incorporate them into the text.

Model Terminology

We appreciate the reviewer's comment regarding our use of the term LLM. At the time of writing, the terms LMM and LVLM lacked consistent definitions and were comparatively much-less used. By referring to our framework as IG-LLM, we intended to better highlight our exploration of inverse graphics as a language task and differentiate our work from those solving coarse semantic-level tasks. Our use of the term LLM is also in line with prior work such as Hong et al., 2023 (3D-LLM). We will add a sentence to the paper to better clarify this.