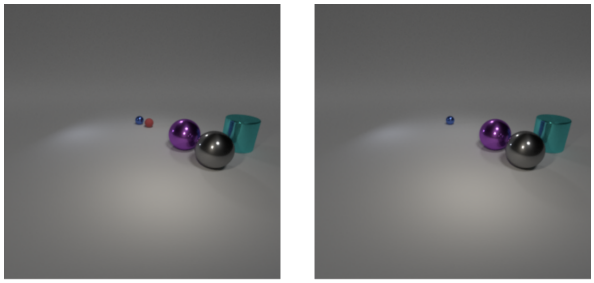
APPENDIX

A. An example from CLEVR-POC

Complete and incomplete scene: Figure 1 is an example of a complete scene and the incomplete scene generated from it by hiding the *small red rubber sphere*.



Complete scene **Incomplete scene**

Figure 1: A complete and incomplete scene from CLEVR-POC

Environment Every scene is generated such that it satisfies the constraints in an environment. The following are the general rules shared by all environments in CLEVR-POC.

```
 1. property(color, gray). property(color, red).
 2. property(color, blue). property(color, green).
 3. property(color, brown). property(color, purple).
 4. property(color, cyan). property(color, yellow).
 5. property(shape, cube). property(shape, cylinder).
 6. property(shape, sphere). property(shape, cone).
 7. property(size, small). property(size, medium).
 8. property(size, large).
 9. property(material, rubber).
    property(material, metal).
10. region(0). region(1). region(2). region(3).
11. right_R(0, 0). right_R(0, 1). right_R(0, 2).
    right_R(0, 3).
12. right_R(1, 1). right_R(1, 3).
13. right_R(2, 0). right_R(2, 1). right_R(2, 2).
    right_R(2, 3).
14. right_R(3, 1). right_R(3, 3).
15. left_R(R1, R2) :- right_R(R2, R1).
16. front_R(0, 0). front_R(0, 1). front_R(0, 2).
    front_R(0, 3).
17. front_R(1, 0). front_R(1, 1). front_R(1, 2).
    front_R(1, 3).
18. front_R(2, 2). front_R(2, 3).
19. front_R(3, 2). front_R(3, 3).
20. behind_R(R1, R2) :- front_R(R2, R1).
21. sameProperty(X1, X2, P) :- hasProperty(X1,P,V),}
22. hasProperty(X2,P,V), X1!=X2.
23. same_color(X,Y):- sameProperty(X, Y, color).
24. same_size(X,Y):- sameProperty(X, Y, size).
25. same_shape(X,Y):- sameProperty(X, Y, shape).
26. same_material(X,Y):- sameProperty(X, Y, material).
```

```
27. 1{hasProperty(X, color, V) :
28.            property(color, V)}1 :-  object(X).
29. 1{hasProperty(X, material, V) :
30.            property(material, V)}1 :- object(X).
31. 1{hasProperty(X, shape, V) :
32.            property(shape, V)}1 :- object(X).
33. 1{hasProperty(X, size, V) :
34.            property(size, V)}1 :- object(X).
35.1{at(X, R): region(R)}1 :- object(X).
36.:- sameProperty(X1, X2, color),
37.   sameProperty(X1, X2, material),
38.   sameProperty(X1, X2, size)},
39.   sameProperty(X1, X2, shape),
40.   object(X1), object(X2), X1!=X2.

41.exceed_region_capacity(R) :-
42.   #count{X: object(X), at(X, R)} >= 4, region(R).
43:- exceed_region_capacity(_).
```

Environment's general rules in natural language:

```
1-9. Objects must have 4 properties. They are color,
shape, size, and material.

1-4. Objects can be in one of the 8 colors. It can
be gray, or red, or blue, or green, or brown,
or purple, or cyan,  or yellow.

5-6. Objects can be in one of the 4 shapes.
It can be cube, or a cylinder, or a sphere or cone.

7-8. Objects can be in one of the 3 sizes.
It can be small, or medium, or large.

9. Objects can be in one of the 2 materials.
It can be rubber, or metal.

10.The scene is divided into 4 regions.
They are named 0, 1, 2, 3.

11. If there are two objects, the first object is
located  in region 0 and the second object is to
the right of the  first object, then the location
of second object is either in region 0, or 1, or
2, or 3.

12. If there are two objects, the first object is
located in region 1 and the second object is to
the right of the first object, then the location
of second object is either in region 1, or 3.

13. If there are two objects, the first object is
located in region 2 and the second object is to
the right of the first object, then the location
of second object is either in region 0, or 1, or
2, or 3.

14. If there are two objects, the first object is
located in region 3 and the second object is to
the right of the first object, then the location
of second object is either in region 1, or 3.
```

```
15. If there are two objects, the first object is
to the right of the second object, then the second
object is to the left of first object.

16. If there are two objects, the first object is
located in region 0 and the second object is in
front of the first object, then the location of
second  object is either  in region 0, or 1, or
2, or 3.

17. If there are two objects, the first object is
located in region 1 and the second object is in
front of the first object, then the location of
second object is either in region 0, or 1, or
2, or 3.

18. If there are two objects, the first object is
located in region 2 and the second object is in
front of the first object, then the location of
second object is either  in region 2, or 3.

19. If there are two objects, the first object is
located in region 3 and the second object is in
front of the first object, then the location of
second  object is either in region 2, or 3.

20. If there are two objects, the first object is
in front of the second object, then the second
object is behind the first object.

27-28. Every object must be assigned exactly one
value for color.

29-30. Every object must be assigned exactly one
value for material.

31-32. Every object must be assigned exactly one
value for shape.

33-34. Every object must be assigned exactly one
value for size.

35. Every object must be assigned exactly one value
for region.

36-40. Two different objects cannot have same values
for all the 4 properties.

41-43. Every region can have at most 3 objects.
```

The following constraints in ASP represent the specific environment to which the scene in Figure 1 belongs to.

```
44. object(0..4).

45. :- object(X), at(X, 0),
                hasProperty(X, size, large).
46. :- object(X), at(X, 0),
                hasProperty(X, shape, cylinder).
47. :- object(X), at(X, 0),
                hasProperty(X, shape, cone).
48. :- object(X), at(X, 1),
                hasProperty(X, size, small).
49. :- object(X), at(X, 1),
                hasProperty(X, shape, cone).
50. :- object(X), at(X, 1),
                hasProperty(X, material, rubber).
51. :- object(X), at(X, 1),
                hasProperty(X, shape, cube).
52. :- object(X), at(X, 2),
                not hasProperty(X, size, medium).
53. :- object(X), at(X, 2),
                not hasProperty(X, material, metal).
54. :- object(X), at(X, 2),
                hasProperty(X, material, rubber).
55. :- object(X), at(X, 2),
                hasProperty(X, shape, sphere).
56. :- object(X), at(X, 2),
                hasProperty(X, shape, cube).
57. :- object(X), at(X, 3),
                hasProperty(X, size, small).
58  :- object(X), at(X, 3),
                not hasProperty(X, material, metal),
59. not hasProperty(X, color, blue).
60. :- #count{X1, X2: sameProperty(X1, X2, shape),
61.     object(X1), object(X2), at(X1, 3), at(X2, 2),
62.     hasProperty(X1, color, yellow),
63.     hasProperty(X2, color, yellow)} >= 4.

64. :- #count{X1, X2: sameProperty(X1, X2, color),
65.     object(X1), object(X2),
66.     at(X1, 0), at(X2, 3)} >= 2.
```

The following is a natural language interpretation of each line of the preceding rules.

```
44. There are 5 objects in the scene.

45. There are no large size objects in region 0.

46. There are no cylinder shape objects in region 0.

47. There are no cone shape objects in region 0.

48. There are no small size objects in region 1.

49. There are no cone shape objects in region 1.

50. There are no rubber material objects in region 1.

51. There are no cube shape objects in region 1.

52. All objects in region 2 have medium size.

53. All objects in region 2 have metal material.

54. There are no rubber material objects in region 2.

55. There are no sphere shape objects in region 2.

56. There are no cube shape objects in region 2.

57. There are no small size objects in region 3.

58-59. All objects in region 3 have either metal
material or blue color.

60-63. There are at most 1 pairs of color yellow
objects with the same shape in regions 3 and 2
together.

64-66. There are at most 0 pairs of objects with the
same color in regions 0 and 3 together.
```

Question: For each given incomplete scene, we generate one question about (any property) of the missing object. The following is the **question in natural language** that is associated with the incomplete scene in Figure 1:

> *There is another red rubber object that is the same shape as the big purple object; what size is it?*

The following is the same **question represented in ASP**:

```
1.  missing(Q)  :-
2.          hasProperty(X,size,Q),
3.          hasProperty(X,material,rubber),
4.          hasProperty(X,color,red),
```

```
5.          hasProperty(Y,color,purple),
6.          hasProperty(Y,size,large),
7.          X!=Y,
8.          same_shape(Y,X).
```

Answer set: The answer set for the above question that satisfies the constraints in the specified environment is:

```
{small, medium}
```

Reasoning Steps: The reasoning involved in deriving the answer set from the question, the incomplete scene, and the constraints in the specified environment is given below.

- Interpreting each line of the question:

```
1. What are the possible values for Q such that:
2. Q is size of the missing object,
3. the missing object's material is rubber,
4. the missing object's color is red,
5. the reference object's color is purple,
6. the reference object's size is large,
7. the missing object is not equal to the
   reference object,
8. the missing object's shape = the
   reference object's shape.
```

- Inferring the missing object's properties:
  from the scene graph: =>

```
8. the reference object's shape is sphere.
9. => the missing object's shape is also sphere.
10. => missing object is a red rubber sphere.
```

- Inferring the missing object's possible **regions** based on the rules listed as the **Environment's constraints**:
  Among the four regions:

```
A red rubber sphere CAN be located at region 0,
as none of the constraints at lines 45-47 is
violated.

A red rubber sphere CAN'T be located at region
1, as it violates the constraint about the
material at line 50.

A red rubber sphere CAN'T be located at region
2, as it violates the constraints at lines 53,
54, and 55.

A red rubber sphere CAN'T be located at region
3, as it violates the constraints at lines 58-59.

=> The missing red rubber sphere is located at
region 0.
```

- Inferring the possible answer set for the property of interest w.r.t the inferred location of the missing object:

```
There are 3 possible values for the size property:
small, medium, large.

The environment constraint at line 45 discards the
large size for region 0.

=> The possible answer set for Q is: small, medium
```

B. Dataset Statistics

Distribution across question types: The type of question asked depends on which attribute of the object is being inquired about. The generation process enables the user to have control over this distribution. For instance, when generating the specific dataset, we established the following criteria: 40% of the questions pertain to the color attribute, another 40% focus on the shape attribute, 10% address the size attribute, and the remaining 10% relate to the material attribute. We made this selection based on the observation that attributes like color and shape encompass a larger set of values (8 values for color and 4 for shape) in comparison to material (which has just two values). Consequently, the solution space for questions centered around color is more extensive than that for material, resulting in a more diverse solution space for the dataset. Figure 3 (a) displays the question type distribution of the dataset generated based on this setting.
Distribution across solutions: Figure 3 (b), (c), (d), and (e) illustrate the distribution of potential solutions for various question types: size, shape, material, and color, respectively. We aim for a balanced distribution, avoiding a situation where the majority of questions lead to the same set of answers. For instance, when a question pertains to the shape of an object, its possible solutions could be one of sphere, cube, sphere, or cone, cube, cylinder, as depicted in Figure 3 (c). Since the possible solutions for questions with query attribute *color* is large (as *color* can take 8 values), the entire space is not listed in Figure 3(e). But, it can be seen that the distribution is not favouring any specific solution.
Distribution across question templates: Figure 2 (a) shows the distribution of questions across different question templates. Six templates present in the original CLEVR dataset are used in CLEVR-POC.
Distribution of query attributes with number of objects in the scene: Figure 2 (b) shows the distribution of questions of a specific type based on the number of objects in the scene.

C. Prompts for Language Model

C.1. Stand-alone GPT-4 to solve CLEVR-POC

The format of prompt provided to GPT-4 when employing it to solve CLEVR-POC is shown below. The prompt contains the task description, the scene description, the constraints or knowledge associated with the scene, the question about the scene and the answer. The prompt contains two such examples.

Task description : You are a helpful assistant that answers questions about hidden objects based on scene description and the constraints in the scene. The scene graph is in json format with the following keys. The key objects contains a list of objects present in the scene. Each object has various attributes like material, color, shape, size, region. The key relationships holds information about the spatial relationships between objects in the scene. It contains sub-fields like "front," "right," "left," "behind," etc., each associated with a list of object indices representing objects that have that specific relationship with another object. For example, relationships["front"][0] refers to the objects that are in front of the object at index 0.
Scene Observed: The following is the scene graph:

```
{'objects': [
 {'material': 'metal',  'color': 'red',
  'size': 'medium',  'region': '0',
  'shape': 'cube'
 },
 {'material': 'metal', 'color': 'gray',
  'size': 'medium', 'region': '3',
  'shape': 'sphere'
 },
 {'material': 'metal', 'color': 'brown',
  'size': 'medium', 'region': '1',
  'shape': 'sphere'
 },
```

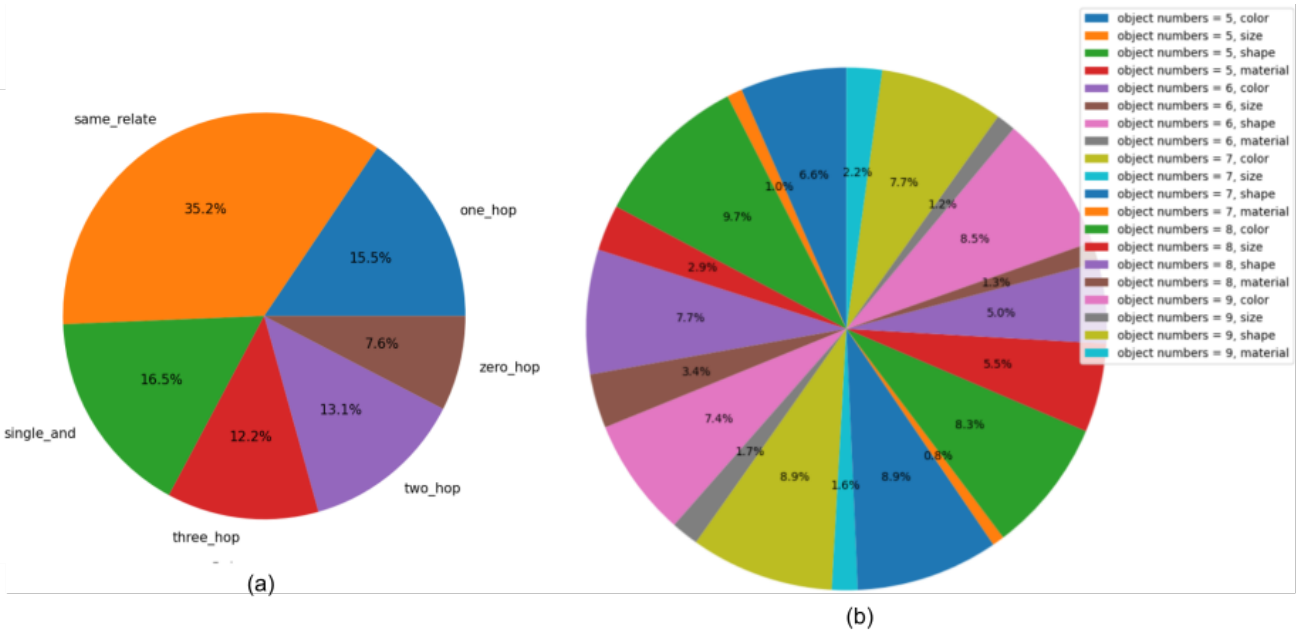


Figure 2: (a) Question templates distribution (b) Distribution of query attributes with object counts between 5 to 9.

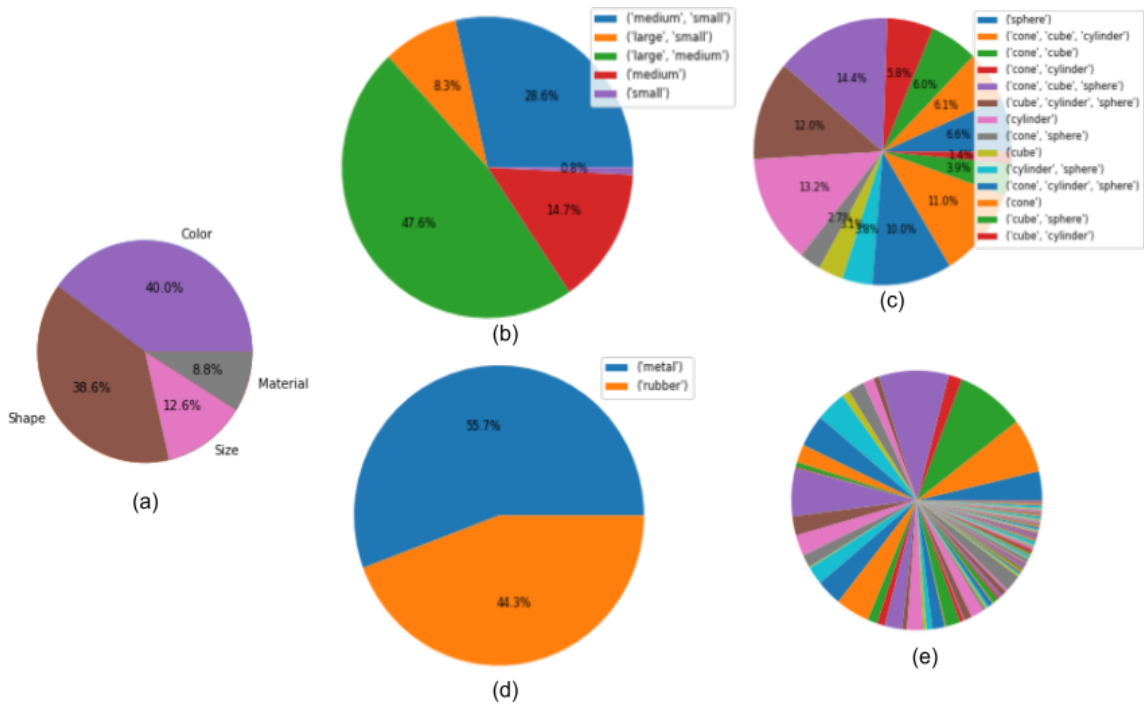


Figure 3: (a) Distribution of question types. (b) Distribution of solutions for questions with query attribute size. (c) Distribution of solutions for questions with query attribute material. (d) Distribution of solutions for questions with query attribute shape. (e) Distribution of solutions for questions with query attribute color. Since the solution space of these questions are larger (> 100), it is not listed here.

```
 {'material': 'rubber', 'color': 'gray',
  'size': 'medium', 'region': '3',
  'shape': 'sphere'
 },
 {'material': 'metal', 'color': 'red',
  'size': 'medium', 'region': '0',
  'shape': 'sphere'
 },
 {'material': 'rubber', 'color': 'red',
  'size': 'medium', 'region': '2',
  'shape': 'sphere'
 }
],
 'relationships':
  {'left': [[4], [0, 2, 4, 5], [0, 4, 5],  [0, 1,
      2, 4, 5], [], [0, 4]],
   'front': [[1, 3, 4, 5], [5], [0, 1, 3, 4, 5],
       [1, 5], [1, 3, 5], []],
   'behind': [[2], [0, 2, 3, 4], [], [0, 2, 4], [0,
       2], [0, 1, 2, 3, 4]],
   'right': [[1, 2, 3, 5], [3], [1, 3], [], [0, 1,
       2, 3, 5], [1, 2, 3]]
  }
}
```

Constraints: The scene contains several visible with one additional that is hidden. Objects must have 4 properties. They are color, shape, size, and material. The scene must conform to the following constraints.

```
Objects can be in one of the 8 colors. It can be
gray, or red, or blue, or green, or brown, or
purple, or cyan, or yellow.

Objects can be in one of the 4 shapes. It can be
cube, or cylinder, or sphere or cone.

Objects can be in one of the 3 sizes. It can be
small, or medium, or large.

Objects can be in one of the 2 materials. It can be
rubber, or metal.

The scene is divided into 4 regions. They are named
0, 1, 2, 3.

If there are two objects, and the first object is
located in region 0 and the second object is to the
right of the first object, then the location of
second object is either in region 0, or 1, or 2, or
3.

If there are two objects, and the first object is
located in region 1 and the second object is to the
right of the first object, then the location of
second object is either in region 1, or 3.

If there are two objects, and the first object is
located in region 2 and the second object is to the
right of the first object, then the location of
second object is either in region 0, or 1, or
2, or 3.

If there are two objects, the first object is
located in region 3 and the second object is
to the right of the first object, then the
location of second object is either in region
1, or 3.

If there are two objects, the first object is
to the right of the second object, then the
second object is to the left of first object.

If there are two objects, the first object is
located in region 0 and the second object is
in front of the first object, then the location
of second object is either in region
0, or 1, or 2, or 3.

If there are two objects, the first object is
located in region 1 and the second object is
in front of the first object, then the location
of second object is either in region
0, or 1, or 2, or 3.

If there are two objects, the first object is
located in region 2 and the second object is
in front of the first object, then the location
of second object is either in region 2, or 3.

If there are two objects, the first object is
located in region 3 and the second object is
in front of the first object, then the location
of second object is either in region 2, or 3.

If there are two objects, the first object is
in front of the second object, then the second
object is behind the first object.

Every object must be assigned exactly one value
for color.

Every object must be assigned exactly one value
for material.

Every object must be assigned exactly one value
for shape.

Every object must be assigned exactly one value
for size.

Every object must be assigned exactly one value
for region.

Two different objects cannot have same values
for all the 4 properties.

Every region can have at most 3 objects.

There are 6 objects in the scene.

There are at least 1 pairs of color red objects
with same size in regions 0 and 2 together.

There are no small size objects in region 0.
```

```
There are no cone shape objects in region 0.

There are no purple color objects in region 0.

There are no blue color objects in region 0.

There are no cylinder shape objects in region 1.

There are no cyan color objects in region 1.

There are no rubber material objects in region 1.

There are at least 1 pairs of material metal
objects with same size in regions 0 and 3 together.

There are no metal material objects in region 2.

There are no large size objects in region 2.

There are at least 1 pairs of size medium
objects with same shape in regions 1 and 3 together
    .

There are no red color objects in region 3.

There are no cube shape objects in region 3.

There are at least 1 pairs of objects with
same material in regions 0 and 1 together.

There are at least 1 pairs of color gray
objects with same size in regions 3 and 2 together.
```

Question: Answer the following question about the hidden object. The solution should satisfy the constraints. The other cylinder that is the same material as the medium red thing is what color?
Answer: ###{Gray}###

C.2. GPT-4 as Question Parser

When we use GPT-4 to parse a question to its ASP equivalent, we give as prompt 28 examples of question in natural language to ASP representation. The prompt with just one Question-ASP pair is shown below.

Task description: You are a helpful assistant that converts questions in English into ASP logic language.
Question: What is the color of cylinder to the right of the blue sphere?
ASP:

```
###
unknown(Q):-hasProperty(X, color, Q),
            hasProperty(X, shape, cylinder),
            hasProperty(X1, color, blue),
            hasProperty(X1, shape, sphere),
            right(X1, X).
###
```