PTR: A Benchmark for Part-based Conceptual, Relational, and Physical Reasoning

Yining Hong *
UCLA

Li Yi
Stanford University

Joshua B. Tenenbaum
MIT BCS, CBMM, CSAIL

Antonio Torralba
MIT CSAIL

Chuang Gan
MIT-IBM Watson AI Lab

Abstract

A critical aspect of human visual perception is the ability to parse visual scenes into individual objects and further into object parts, forming part-whole hierarchies. Such composite structures could induce a rich set of semantic concepts and relations, thus playing an important role in the interpretation and organization of visual signals as well as for the generalization of visual perception and reasoning. However, existing visual reasoning benchmarks mostly focus on objects rather than parts. Visual reasoning based on the full part-whole hierarchy is much more challenging than object-centric reasoning due to finer-grained concepts, richer geometry relations, and more complex physics. Therefore, to better serve for part-based conceptual, relational and physical reasoning, we introduce a new large-scale diagnostic visual reasoning dataset named PTR. PTR contains around 70k RGBD synthetic images with ground truth object and part level annotations regarding semantic instance segmentation, color attributes, spatial and geometric relationships, and certain physical properties such as stability. These images are paired with 700k machine-generated questions covering various types of reasoning types, making them a good testbed for visual reasoning models. We examine several state-of-the-art visual reasoning models on this dataset and observe that they still make many surprising mistakes in situations where humans can easily infer the correct answer. We believe this dataset will open up new opportunities for part-based reasoning. PTR dataset and baseline models are publicly available [2].

1 Introduction

A long-standing challenge in the field of artificial intelligence is to enable machines to reason and answer questions about visual scenes. Several datasets [64, 16, 29, 49, 57] have been proposed to tackle this challenge. They mostly focus on object-level features without emphasizing much on detailed part-level understanding. However, there is strong psychological evidence that human beings parse visual scenes into part-whole hierarchies (*e.g.,* from a scene to the objects, from an object to its parts), which are currently missing from existing datasets.

Inspired by Aristotle's quote "the whole is greater than the sum of its parts", there has been a long history of research in psychology concerning how parts and wholes are related, beginning with the Gestalt psychologists [47]. There have also been recent efforts on how to represent such part-whole hierarchies using neural networks [19]. Introducing part-whole hierarchies into visual reasoning

*The work was done while Yining Hong was a research intern at MIT-IBM Watson AI Lab.

[2]Project page: http://ptr.csail.mit.edu/

35th Conference on Neural Information Processing Systems (NeurIPS 2021), virtual.

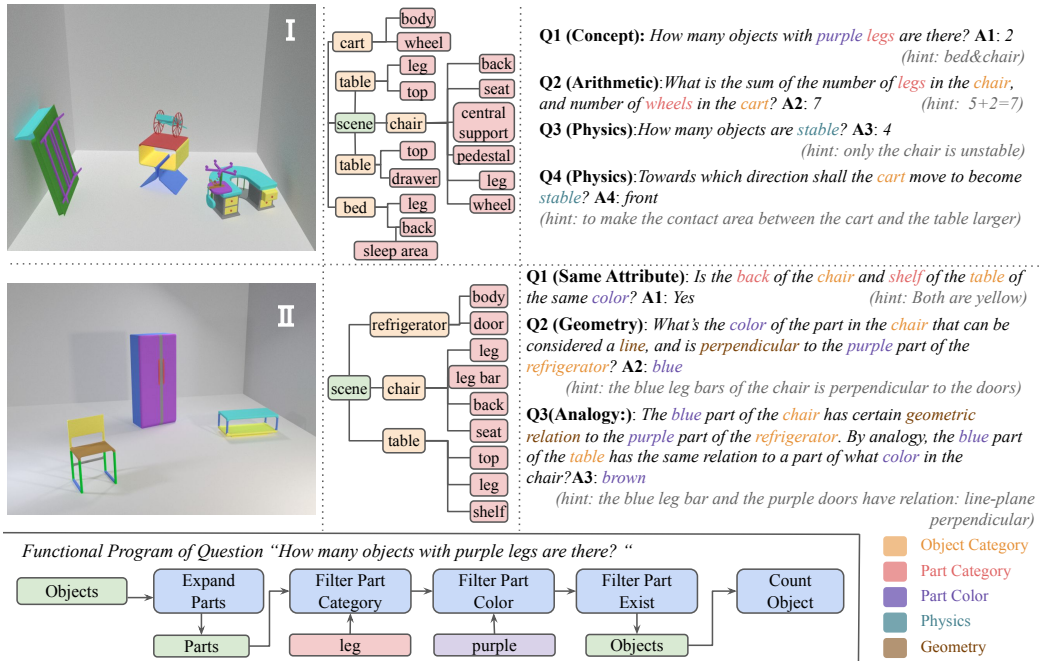


Figure 1: Exemplar scene graphs, questions, and programs of our Part Reasoning (PTR) dataset. PTR contains five question types: concept, relation, analogy, arithmetic and physics. Top left shows scenes from our dataset, paired with hierarchical scene graphs in the middle. Top right presents questions, answers and hints to the answers. A functional program of a question is shown in the bottom.

brings two unique challenges. The first is how to discriminate objects using part-level attributes. Unlike previous works which refer to objects based on holistic attributes such as object category, the specific ontology nature of objects and their parts requires a detailed study. The second is how to leverage part-level attributes to interconnect objects. Objects of various categories are interrelated via some common parts (*e.g.*, leg, central support, pedestal). The reasoning performed on one category should thus be easily generalized to reasoning on unseen categories with shared parts. Human beings are endowed with such modular but interconnected perceptual systems. They can effortlessly discriminate between the two tables in Figure 1 [I] based on the differences in the tops and drawers, as well as observe a connection between the bed and chair due to the similarity of legs. It remains to be explored whether machines have the same hierarchical perceptual capabilities.

Based on the highly composite part-whole hierarchies, visual reasoning tasks can be richer, more complex, as well as more challenging. First, the introduction of parts enriches the diversity of both visual perception and question understanding. Apart from basic object-level properties such as color and category, the visual scenes and natural language questions extend upon the properties of object parts, the composition of which makes the image-question pairs more distinctive. Second, the relations between parts go beyond simple spatial relations. Parts can often be approximated as oriented geometric primitives (*e.g.,* line, plane), and there naturally exist abundant geometric relations between these primitives such as "perpendicular" and "parallel" (*e.g.,* Q2 in Figure 1[II]). Analogical reasoning can also be established given such relationships. Third, reasoning on parts enables the understanding of implicit properties of objects, such as physics. For example, the arrangements of the parts and their geometric relations affect the stability of the objects (*e.g.,* Q3, Q4 in Figure 1[I]).

In this paper, we present a large-scale ParT Reasoning dataset, or PTR for short, a benchmark for part-based conceptual, relational and physical reasoning. It includes $\sim$ 70k scenes paired with 700k questions containing part-whole structures. We include over 10k objects from the PartNet [45] dataset across five object categories (chair, table, bed, refrigerator, cart) with rich geometric and structural variations to construct our scenes. Our dataset consists of five types of questions: concept, geometry, analogy, arithmetic, and physics. We provide scene graph annotations including the ground-truth locations, segmentations, and properties for all objects and parts, as well as all the object-object

Dataset	3D	Objects	Parts	Relation	Diagnostic	Geometry	Analogy	Physics
VQA [3]	✓	✓	✗	✓	✗	✗	✗	✗
VCR [61]	✓	✓	✗	✓	✗	✗	✗	✗
GQA [28]	✓	✓	✗	✓	✓	✗	✗	✗
CLEVR [29]	✓	✓	✗	✓	✓	✗	✗	✗
RAVEN [62]	✗	✓	✓	✓	✓	✗	✓	✗
PTR (ours)	✓	✓	✓	✓	✓	✓	✓	✓

Table 1: Comparison between PTR and other visual reasoning benchmarks.

and part-part relationships for model diagnostic. We also provide functional programs paired with questions.

We analyze a suite of state-of-the-art visual reasoning models on the PTR dataset and find that they all struggle with it, especially in relational, analogical, and physical reasoning. One oracle neural symbolic model performs better than other models but highly relies on additional supervision such as object and part masks and visual attributes from simulations. Also, the performances of all the models are inferior to human performance by a large margin, suggesting that there's still a long way to go to equip machines with human-like hierarchical perceptual and reasoning abilities.

2 Related Work

2.1 Visual Reasoning

To assess machines' ability on understanding and reasoning over vision and language, a wide variety of benchmarks have been created over the recent years [29, 3, 28, 16, 61, 12, 64, 36]. In terms of data input, our dataset mostly resembles the CLEVR dataset [29], which is also a diagnostic dataset of synthetic images, template-based questions and compositional programs. However, the objects in CLEVR's scenes are of simple shapes and do not contain parts to form complex logical chains of reasoning. The VQA dataset [3] contains large-scale crowd-sourced real images and human-generated questions. Since it's not a fully-controlled synthetic dataset, it shows great inherent bias. Visual Commonsense Reasoning (VCR) [61] focuses on reasoning on commonsense knowledge. The GQA dataset [28] pairs real images with synthetic compositional questions. However, none of these datasets provide part-based dense annotations and questions. RAVEN [62] associates vision with structural, relational, and analogical reasoning in a hierarchical representation. The dataset asks models to choose an image according to analogy. However, their images are rather simple, mainly consisting of 2D shapes like circles and triangles, and they do not pair images with natural language questions.

Although numerous visual reasoning datasets have been proposed, the reasoning chain typically terminates at the object level for these datasets, due to the mere reference to objects based on holistic features or spatial relationships. Also, while previous datasets emphasize reasoning based on spatial relationships, humans possess diverse reasoning abilities, including analogy [62], geometry[48, 40], arithmetic[63, 20, 21], and physics [13]. These aspects can be effortlessly incorporated into visual reasoning to constitute a benchmark that further approaches humans' intellectual level. Thus, we create a new visual-reasoning dataset that focuses on multiple aspects of reasoning on part-whole relations. We compare our dataset with previous visual reasoning datasets in Table 1.

A large number of visual reasoning models have been proposed. Specifically, MAC [27] combined multi-modal attention for compositional reasoning. LCGN [23] built contextualized representations for objects to support relational reasoning. These methods, though embedded in neural networks, model the reasoning process implicitly. Neural-symbolic methods [56, 42, 7, 38] explicitly perform symbolic reasoning on the objects representations and language representations. Recently, methods based on transformers [33] have achieved remarkable performance on visual reasoning tasks by utilizing end-to-end object detector. In this paper, we conduct experiments and assess strengths and weaknesses of these baseline models on part-based visual reasoning.

2.2 Part-based 3D Understanding

Understanding 3D parts has been a long-standing problem in computer vision and graphics. It is important due to several reasons. Firstly, parts and their arrangements that describe the detailed object geometry and structure are helpful for object discrimination purposes [9, 1]. Secondly, parts are usually action handles and their forms could reveal the object functionality [34, 43, 24], therefore crucial for interacting with objects. Thirdly, functionally related object parts usually share similar forms, making parts suitable for connecting different objects, facilitating knowledge transfer and promoting generalizable learning [17, 41]. Finally, parts are often commonly used for 3D generation and content editing purposes [11, 44, 46].

To facilitate part-based 3D understanding, various part-centric 3D datasets have been proposed in the literature. ShapeNet part dataset [58] annotates 31,963 shapes from ShapeNet [4] with semantic part segmentation covering 16 categories. And each shape is annotated with 2 to 5 parts. PartNet [45] further increases the granularity of parts, contributing 573,585 finegrained part instance annotations for 26,671 shapes across 24 object categories. Going beyond semantic understanding, several datasets also focus on the functionality and articulation aspects of parts [54, 43, 24, 25, 52]. However, most of these datasets target at part identification rather than part-based conceptual, relational and physical reasoning. Chang et al. [5] collected 27,477 part instances from 2,278 models covering 90 object categories, with a focus of studying conceptual part-based reasoning. ShapeGlot [1] presents an object reference dataset with part-level conceptual reasoning implicitly considered. Want et al. [51] proposes PartNet-Chairs, which contain questions about parts of chairs. These datasets are restricted to conceptual reasoning while our PTR has a broader scope covering ampler reasoning types.

Restricted by the annotations in existing large-scale datasets, previous part-based 3D understanding works mainly focus on identifying parts [15, 30, 26, 55, 32, 58] and part relations [6, 10, 31, 35, 60, 50] for individual 3D objects. Several methods also consider a more complex hierarchical structure of 3D parts [59, 44, 37, 53, 14, 22, 8]. Moreover, these works usually assume a strong categorical prior and tend to have poor generalizability across categories. However, we human beings can easily do more than just part identification or single-object part relationship understanding for objects from known categories. We can reason the relationship between parts from different object instances, make analogies, conduct mental arithmetic, and even infer the physical realism or stability. These tasks are not well supported by current datasets. Therefore we present PTR to close this gap and to better support the study of these human-like common sense.

3 The PTR Dataset

The PTR dataset aims to assess machines' ability to perform part-based visual reasoning. We carefully design this dataset in a fully-controlled synthetic environment and enable model diagnostics on this complex reasoning task. Each image has ground-truth annotations regarding the locations, orientations and attributes of objects and parts, as well as the object masks and part masks. We also provide depth images, camera locations and orientations to facilitate the possible use of 3D models on our dataset in the future. For reasoning on geometric relations of parts, we include the geometric information of a part indicating whether it can be considered as a geometric line or plane, and the line equation or plane equation. For physical reasoning, we include the stability information of each object. The images are paired with five types of questions: concept, relation, analogy, arithmetic and physics. The questions have associated functional programs.

3.1 Image Generation

PTR includes approximately 52k images for training, 9k for validation and 10k for testing. The images are rendered via Blender[3]. To get the physical information, we apply Bullet[4] to simulate the future motion traces of each object.

Scene, Objects and Parts. The PTR universe contains five categories of objects (chair, table, bed, refrigerator and cart) from the PartNet dataset. The reasons why we choose these categories are as follows: 1) These are commonly seen objects in real life scenes; 2)The objects share a lot of generic

[3]https://www.blender.org/

[4]https://pybullet.org/wordpress/

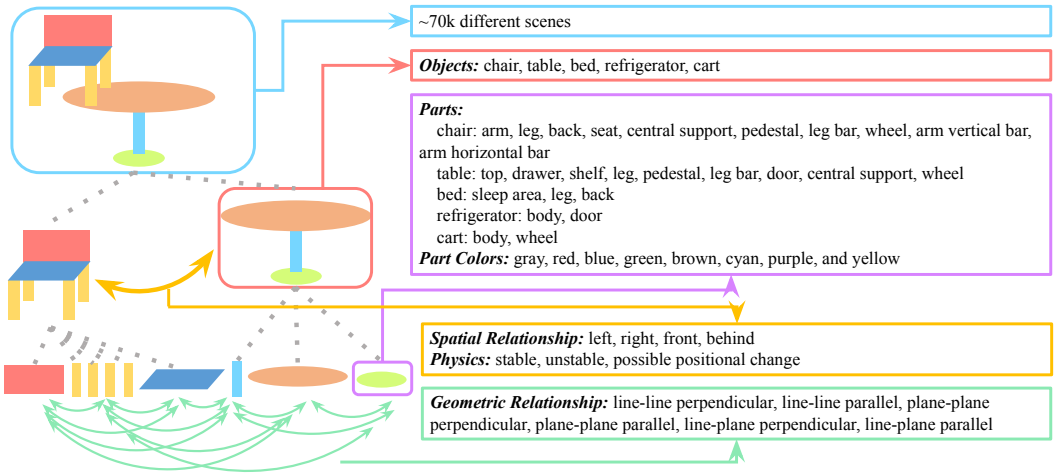


Figure 2: An overview of the scene structure and concepts in the PTR dataset. The PTR universe contains five object categories, each consisting of a set of part categories. Objects have spatial relationships and parts have geometric relationships. Physical properties are also defined on objects.

parts (*e.g.,* leg, leg bar, central support, pedestal, door, wheel, *etc.*); 3)The objects have rich physical configurations depending on the constitution of the parts. Each object can have an arbitrary number of parts. Each part takes one of eight colors (gray, red, blue, green, brown, cyan, purple, and yellow). The specific concepts used in the PTR dataset are listed in Figure 2. The distribution of the number of parts in the scenes is shown in Figure 3c.

When constructing the scene, we first place a floor and three walls of random texture to enrich geometric and physical information. We add jitters to the locations and orientations of the camera and the lamps. Every scene has three to six objects. To place an object, we ensure that no objects overlap (except for physical scenes which have objects stacking). We also ensure that the objects are of proper orientations so that no much burden would be placed on object detection. The training dataset and test dataset has no shared 3D object models (*i.e,*, the PartNet 3D models are not shared across splits, but the semantics are shared).

The physical scenes contain objects that are tilted, leaned towards walls, or stacked onto other objects. We use the Bullet physics engine to simulate physical effects. We calculate the change of object locations and orientations along time steps to determine the stability of an object. If the change of location and orientation is within a threshold value, an object can be considered stable. For unstable objects, we move the objects in four directions (front, behind, left, right) around their original location, to account for possible changes for the objects to become stable. To avoid missing such possible changes, for each direction we define four moving distances to place the objects.

Relationships PTR has three types of relationships: spatial relationships of objects, geometric relationships of parts, and same-attribute relationships of both objects and parts.

Objects are spatially related via four relationships (left, right, in front of, behind). The spatial relationships are dependent upon the camera viewpoint.

Parts have geometric relations if they, with negligible thickness, can be considered as 1D lines or 2D planes (*e.g.,* the legs and leg bars of Figure 1 [II] can be treated as lines, and the doors can be treated as planes). To detect such geometric primitives, for each part we randomly sample two points for lines, or three points for planes from the raw 3D data, for 1000 times. Each time we derive a line equation or plane equation from the sampled points. We ensure that distances between the points are sufficient enough with regard to the size of the part to reduce noise. The final line or plane equation is then approximated by averaging the equations. We then further sample 2500 points from the 3D data to see if the majority of points conform to the equation. If not, the part cannot be considered as a line or plane. The same-attribute relationships involve same-category relationship between objects, as well as same-category and same-color relationships between parts.

3.2 Question Generation

We pair each image with machine-generated questions coupled with executable functional programs. PTR contains approximately 520k questions for training, 90k for validation and 100k for testing. All questions are open-ended and can be answered with a single word. Sample questions and detailed distributions can be found in Figure 1 and Figure 3.

Concept Conceptual questions evaluate a model's capability to understand and reason about basic part-whole relations. The reasoning tasks are grounded within the compositional space of both object-level and part-level properties. Unlike previous visual reasoning datasets which refer to an object based on the object's holistic attributes, PTR distinguishes objects by the existence of parts and the attributes of parts. To shed light on the interconnected nature of part-whole hierarchies, there are also questions that do not specify any object category but only the common part categories (*e.g.,* Q1 in Figure 1 [I] makes a connection between bed and chair by asking about the objects with legs). Conceptual questions contain multiple sub-types including: *query object category*, *query part category*, *query part color*, *count object*, *exist object* and *count part*.

Relation Relational questions query about the spatial relationships of objects, geometric relationships of parts, as well as same attribute relationships of objects and parts. Upon querying about the spatial relationships, the questions take the six sub-types of the conceptual questions, except that objects are also filtered using spatial information.

As for geometric relationships, we first detect a part that can be considered as a line or a plane, refer to the part using color, and then query the existence/count/color of parts in another object that has certain geometric relation to this part. There are in total six types of geometric relationships, which are listed in Figure 2. Q2 in Figure 1 [II] is an example of geometric questions. By considering the blue leg bars as lines and the purple doors as planes, we can easily examine that the two parts are perpendicular.

The same attribute relationships take three forms. The first one asks the model to compare two attributes of two objects/parts and return a "yes/no" answer (*e.g.,*, Q1 of Figure 1 [II]). The second refers to an object with the "same-category" relation to another object. The third detects whether an object has the same part attributes (*e.g.,* same color of legs) as another object.

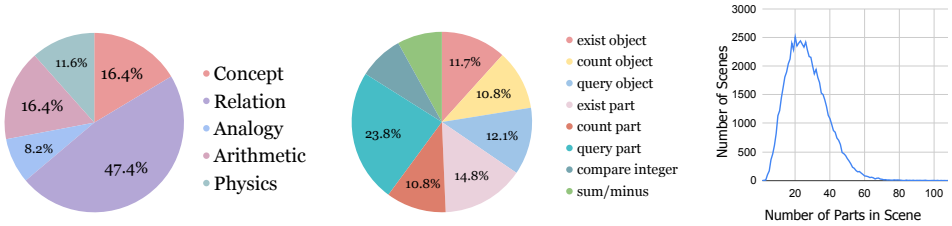
Analogy Analogical questions query analogy on spatial relationships and geometric relationships. Specifically, given objects or parts A, B, C where B has a certain relation to A, the model is required to select an object or part D that has the same relation to C. The question may query about the existence/count/attribute of D. An exemplar question is shown in Figure 1 [II] Q3. In this question, A is the leg bars of the chair, B is the door of the refrigerator. We find that the two parts have the "line-plane perpendicular" relationship. C is the legs of the table, and the question queries about the color of D, such that C and D have the same relation as A and B.

Arithmetic Arithmetic questions take the quantities of two types of parts in two objects, and apply two types of operations to the quantities. The first type is "compare integer". The model is asked to predict whether the two quantities follow "equal/greater than/less than" relationships and return a "yes/no" answer. The other type is "sum/minus"", where the sum or the difference of two numbers is queried (*e.g.,* Figure 1 [I] Q2). We ensure that the answer is no greater than 10 or less than 0 to facilitate the training of neural models.

Physics Physical questions query about the stability of objects. For example, Q3 in Figure 1 [I] asks how many objects are stable. If an object is unstable, further questions can be asked concerning the possible changes to the location of the object to make it stable, as in Figure 1 [I] Q4.

Program In PTR, each question can be parsed into a hierarchical tree-structured functional program, as shown in the functional programs in Figure 1. Different from previous datasets, PTR has three key functions: *expand parts*, *filter part exist* and *filter part count*, which enable the flexible transition between object-level reasoning and part-level reasoning. When the reasoning procedure switches from object level to part level, *expand parts* takes all the objects, and returns the parts of the objects. *filter part exist* filters objects by examining whether certain parts exist in the objects, and *filter part count* filters objects by the number of certain parts in objects. A list of program modules can be found in the Supplementary Material.

Bias control Questions are generated from pre-defined templates, and transformed into natural language questions with associated object-level and part-level attributes from the scene. We manually



(a) Question Semantics (b) Question Reasoning Types (c) Number of Parts

Figure 3: Analysis on question types and distributions.

define 58 templates for question generation. We sort the question types and templates each time we generate a question to ensure a balanced question distribution. To avoid bias, we force a flat answer distribution for each question type by rejection sampling. Specifically, when sampling a question-answer pair, if the count of the answer exceeds the median count of all answers in that type by a margin, we discard the question-answer pair.

4 Baseline Evaluations

4.1 Baseline Models

We mainly evaluate three kinds of state-of-the-art visual reasoning methods, including heuristic models, end-to-end neural networks, and neural-symbolic models.

Q-type (Rand.), Q-type (Freq.): These are language-only baselines that examine the bias in the dataset. Q-type (Rand.) uniformly samples an answer from the answer set given a specific question type. Q-type (Freq.) selects the most frequent training set answer for each question type.

UP-DOWN: We extract part-centric features in an unsupervised manner using Slot Attention[39], and apply bottom-up and top-down attention [2] on the part-centric features.

CNN+LSTM The question is encoded by the final hidden states from a Long Short Term Memory network (LSTM) and the image is encoded using features from a convolutional network (CNN). The features are concatenated and fed to an MLP to predict the final answer. This is a simple baseline that examines how vanilla neural networks perform on PTR.

MAC, MAC(P) ([27]) MAC utilizes a novel Memory, Attention and Composition cell to perform iterative reasoning process. The image features and question features are incorporated via a joint attention mechanism. This is a strong baseline achieving state-of-the-art performance on CLEVR without any supervision on vision and language. To boost part-level reasoning, we construct part-aware features by adding the segmentation masks of all parts and their attributes to the original features, denoted as MAC(P).

LCGN ([23]) The objects are represented as nodes in the graph network, and are described by a context-aware representation from related objects through iterative message passing conditioned on the textual input.

MDETR ([33]) This model fuses the two modalities at an early stage, leveraging an end-to-end modular detector to boost downstream tasks like visual question answering. It uses ground-truth bounding boxes of parts to train the detector.

NS-VQA ([56]) The visual recognition and language understanding are performed by neural networks, which are passed to a symbolic execution module to get the final results. We treat this baseline as an oracle model where we use the annotations of both vision (ground-truth masks) and language (ground-truth programs).

Implementation Details We use an ImageNet-pretrained ResNet-101 to extract $14 \times 14 \times 1024$ feature maps for MAC, MAC(P) and LCGN. For CNN-LSTM, we use the 2048-dimensional feature from the last pooling layer. The setup of MDETR is the same as the original paper with ResNet-101 as backbone. We first train only the task of part detection for 30 epochs, and then train the full PTR with question answering loss.

For NS-VQA, we use Mask R-CNN [18] to generate segmentation proposals of objects and parts, respectively. The Mask R-CNN is trained on 20% of the training data annotated with ground-truth masks for 30,000 iterations. We do not include labels of categories and attributes when training segmentation. We extract the categories and attributes of objects and parts using attribute networks (ResNet-34). The extraction works in a top-down, bottom-up fashion. First, the object proposals are sent to attribute networks to propose object categories. They are then paired with contextual information (*i.e.,* the original images), to produce pose and physics information. Second, the part proposals are augmented with top-down object proposals and categories and sent to the attribute networks, to predict the part categories and colors separately. We also output the probability of a part being a line or a plane, and the orientation (parameters of the line/plane equation) of the geometric parts. Finally, in a bottom-up fashion, if parts cover a region that is not covered by object segmentations, we reconstruct objects by taking the summed area of the parts, and label the objects by taking into account the labels of parts. The program generator of NS-VQA is trained on 1,000 question-program pairs. The part segmentations and attributes are also used in MAC(P).

Question Type	Model	Rand.	Freq.	UP-DOWN	C-LSTM	LCGN	MAC	MAC(P)	MDETR	NS-VQA	Human
Concept	query-object	20.0	26.5	71.8	73.3	76.4	76.8	80.5	79.8	**96.7**	95.5
	exist-object	50.1	52.4	72.6	73.9	77.1	74.8	76.5	78.2	**91.5**	97.8
	count-object	9.8	19.8	50.0	58.9	68.5	63.3	64.5	67.9	**80.8**	95.4
	query-part	8.9	10.9	34.4	35.7	31.5	48.2	49.4	52.3	**71.8**	89.1
	count-part	10.0	20.1	45.5	46.8	50.4	49.6	50.9	51.2	**60.6**	82.9
Relation	spatial	11.5	18.2	40.6	47.2	51.2	54.9	59.3	57.6	**67.1**	85.7
	geometric	40.1	41.7	48.6	57.9	58.6	64.8	65.6	**67.1**	55.3	76.4
	same-attribute	36.1	39.5	47.3	52.9	54.8	57.6	58.0	61.4	**62.5**	94.2
Analogy	spatial	26.1	32.9	19.2	51.0	54.3	53.4	53.3	58.5	**68.7**	84.2
	geometric	12.1	26.7	22.2	26.7	23.3	23.3	30.0	**34.3**	32.5	73.7
Arithmetic	compare-integer	49.7	51.7	69.7	74.0	74.5	75.0	**75.3**	72.0	67.1	83.3
	sum-minus	10.0	19.8	26.8	31.1	35.0	34.2	34.8	36.2	**43.4**	75.6
Physics	stability	37.9	43.5	50.5	56.3	60.7	59.2	**61.5**	58.0	41.2	74.0
	possible-change	40.6	48.1	53.5	54.1	57.2	**57.8**	56.8	53.8	52.0	79.8
All		28.4	33.5	45.4	52.9	55.4	57.5	58.9	59.8	**62.1**	84.8
Supervision		/	/	A	A	A	A	A,M,S	A,M,T	M,S,P	

Table 2: Test accuracies on the PTR dataset. For the supervision types, "A" denotes the answers, "M" denotes the segmentation masks, "S" denotes the semantics of objects and parts, "T" denotes the spans of tokens associated with masks, and "P" denotes the programs of the questions. **Though humans achieve an 84.8% accuracy, SOTA visual reasoning models struggle with it**.

4.2 Result Analysis

We summarize the performances for each question type of baseline models in Table 2. All models are trained on the training set until convergence, tuned on the validation set, and evaluated on the test set.

Overall Result From the table, we can see that the neural-symbolic model, NS-VQA, outperforms neural models by a large margin. This is credited to the extra supervision of both vision (ground-truth part masks and attributes) and language (ground-truth programs). Still, it cannot achieve nearly perfect performances in basic conceptual problems due to the difficulty in perception module (*i.e.,* mask-rcnn and attribute net). MDETR stands out from all the neural models. However, two additional ingredients are required to train MDETR: the bounding boxes of all the parts, and the alignment between vision and language. We further observe that augmenting MAC with part segmentations and attributes boost the accuracy. The results suggest that currently, extra supervision is crucial for both neural models and neural-symbolic models to perform compositional reasoning. On the contrary, if the part segmentations and representations are extracted in an unsupervised manner (*e.g.,* UP-DOWN), the results are much inferior to the results of other baseline models. We also include the unsupervised segmentation results by Slot Attention [39] in the Supplementary Material. The results suggest that our benchmark can also be served as a nature testbed for unsupervised part detection.

All baseline models have unsatisfactory performances compared to human evaluation, especially in question types that require specific reasoning skills, such as analogy, physics and arithmetic.

Question Types We observe a variance of performances across the wide spectrum of reasoning tasks. Starting with physics problems, we find that although NS-VQA has great performances for conceptual

questions, its performance drops drastically when it comes to physical problems, worse than all neural baselines by 15% to 20%. The same phenomenon can be observed in the geometry-type problems, where NS-VQA performs worse than most of the neural baselines. This indicates that the attribute nets in current neural-symbolic models can only learn explicit attributes while struggling with implicit high-level attributes such as physics and geometry. MAC(P) is superior to MAC in "stability" problems, indicating that the physical properties of objects are linked to the composition of parts.

As for tasks that focus on part-level reasoning, models augmented with part-level representations (MAC(P), MDETR, NS-VQA) have better performances than others. Specifically, MDETR excels in geometric problems, and NS-VQA manages to achieve good performances for geometric analogy problems while other models fail. MAC(P) outperforms MAC in query-part, count-part and geometric questions, showing that part-based representations are essential for reasoning on part-level attributes and relationships.

4.3 Data Efficiency

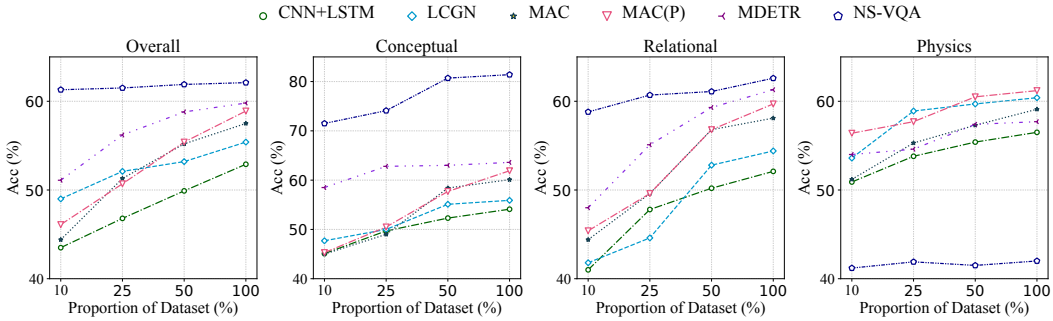


Figure 4: Compared results on data efficiency of different baseline models.

We compare different baselines on a systematic study on data efficiency, which is presented in Figure 4. Specifically, we train the model with 10%, 25%, 50% and 100% randomly chosen training images paired with questions, and evaluate the model on the original testing dataset. For NS-VQA, we use the corresponding percentage of annotated data for part detection and question parsing.

From Figure 4, we find that the neuro-symbolic model remains good performances across all splits. This might be the consequence of the disentanglement of perception module and reasoning module. LCGN is data-efficient, achieving $\sim 49\%$ accuracy with only 10% data, but does not show large improvement with more data. This might be due to the context-aware representations which provide more information with fewer data. All the models improve slightly on the physics category, which suggests that the underperformance on physics problems is rarely caused by insufficient data. Increasing the amount of data boosts the performances in the conceptual questions most, indicating that data is important for learning part-whole concepts.

4.4 Cross-Category Generalization

To examine machines' ability of interconnecting objects of different categories via common parts (*i.e.,* generalizing to unseen categories based on seen categories), we curate a subset from the original dataset. The curated training set has only three object categories: chair, refrigerator, and cart. The testing set contains objects of all five categories, and questions asking about the common parts of the categories (*e.g.,*, both the table and the refrigerator have doors). The cross-category training set has about 15% data of the original training set. For fair comparison, we extract approximately the same amount of data in the training set that have all five categories in the scenes, denoted as "original" in Figure 5.

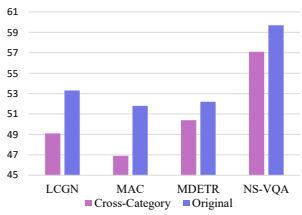


Figure 5: Results on cross-category generalization.

Figure 5 summarizes the cross-category generalization results. From
the figure, we can see that NS-VQA shows great generalization ability, mainly because the part detection and part attribute extraction trained on one category can be easily transferred to another with similar part appearances. Of all the neural models, MDETR has the best performance, which is largely due to modular detection. LCGN also achieves satisfying performance. However, the cross-category accuracy for LCGN is much lower than the original accuracy, suggesting that the good performance is due to data efficiency. MAC(P) has the same problem. Although we augment image features with part segmentations, it does not show good generalization ability given by parts. One explanation is that choosing the right form of part features is crucial. Explicit mechanisms as in MDETR and NS-VQA can lead to better performances and generalization ability.

4.5 Ablative Studies of NS-VQA

We further eliminate perception difficulty, and investigate the challenges of our benchmark if ground-truth part-segmentations are given. We experiment with two settings: 1) Replacing masks output by Mask-RCNN with ground-truth masks (instance segmentations given); 2) Based on 1), replacing attributes output by attribute networks with ground-truth attributes (both instance and semantic segmentations given). The results are shown in Figure 6.

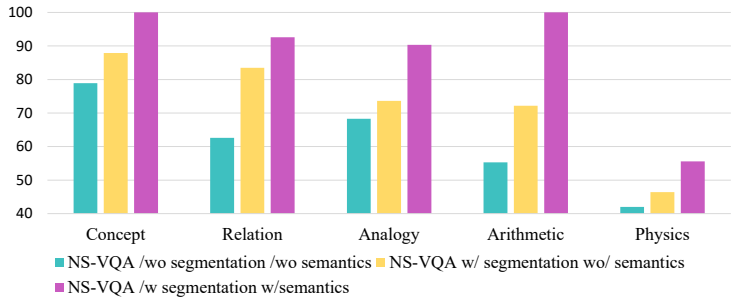


Figure 6: Ablative Studies of NS-VQA model

We can see that given all the ground-truth segmentations and semantics, NS-VQA can achieve perfect results in conceptual and arithmetic problems. However, it still performs poorly in geometric and physical problems. Given only segmentations, NS-VQA has a much lower accuracy than NS-VQA with semantics, showing that existing attribute network architectures also fall short in predicting semantics on our dataset.

5 Conclusion and Future Work

In this paper, we propose PTR, a novel large-scale benchmark that emphasizes visual reasoning on part-whole hierarchies. We propose several challenging question types for PTR, including: concept, relation, analogy, arithmetic and physics. We describe the detailed dataset creation procedure. We further conduct experiments on state-of-the-art visual reasoning models on PTR. Experimental results show that current visual reasoning models underperform humans in part-based reasoning tasks by a large margin. We believe this benchmark will open up opportunities for building and testing a new family of visual reasoning models.

For future work, we can leverage the depth image in PTR and experiment with RGBD-based visual perception models. Also, results show that current models fail to represent part-whole hierarchies delicately, encouraging the design of novel neural networks that specializes in this aspect (*e.g.,* Hinton [19] proposes some feasible ideas.) Last but not least, all models have bad performances in physical reasoning. One promising direction is to integrate physics engines with visual reasoning models to boost performances.

Acknowledgments and Disclosure of Funding

This work was supported by MIT-IBM Watson AI Lab and its member company Nexplore, ONR MURI (N00014-13-1-0333), DARPA Machine Common Sense program, DSTA, ONR (N00014-18-1-2847) and Mitsubishi Electric.

References

[1] Panos Achlioptas, Judy Fan, Robert Hawkins, Noah Goodman, and Leonidas J Guibas. Shapeglot: Learning language for shape differentiation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8938–8947, 2019. 4

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6077–6086, 2018. 7

[3] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *ICCV*, 2015. 3

[4] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 4

[5] Angel X Chang, Rishi Mago, Pranav Krishna, Manolis Savva, and Christiane Fellbaum. Linking wordnet to 3d shapes. In *Global WordNet Conference*, 2018. 4

[6] Siddhartha Chaudhuri, Evangelos Kalogerakis, Leonidas Guibas, and Vladlen Koltun. Probabilistic reasoning for assembly-based 3d modeling. In *ACM SIGGRAPH 2011 papers*, pages 1–10. 2011. 4

[7] Zhenfang Chen, Jiayuan Mao, Jiajun Wu, Kwan-Yee Kenneth Wong, Joshua B Tenenbaum, and Chuang Gan. Grounding physical concepts of objects and events through dynamic visual reasoning. *ICLR*, 2021. 3

[8] Fei Deng, Zhuo Zhi, Donghun Lee, and Sungjin Ahn. Generative scene graph networks. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=RmcPm9m3tnk. 4

[9] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2009. 4

[10] Noa Fish, Melinos Averkiou, Oliver Van Kaick, Olga Sorkine-Hornung, Daniel Cohen-Or, and Niloy J Mitra. Meta-representation of shape families. *ACM Transactions on Graphics (TOG)*, 33(4):1–11, 2014. 4

[11] Thomas Funkhouser, Michael Kazhdan, Philip Shilane, Patrick Min, William Kiefer, Ayellet Tal, Szymon Rusinkiewicz, and David Dobkin. Modeling by example. *ACM transactions on graphics (TOG)*, 23(3):652–663, 2004. 4

[12] Chuang Gan, Yandong Li, Haoxiang Li, Chen Sun, and Boqing Gong. VQS: Linking segmentations to questions and answers for supervised attention in vqa and question-focused semantic segmentation. In *ICCV*, pages 1811–1820, 2017. 3

[13] Chuang Gan, Jeremy Schwartz, Seth Alter, Martin Schrimpf, James Traer, Julian De Freitas, Jonas Kubilius, Abhishek Bhandwaldar, Nick Haber, Megumi Sano, et al. Threedworld: A platform for interactive multi-modal physical simulation. *NeurIPS*, 2021. 3

[14] Lin Gao, Jie Yang, Tong Wu, Yu-Jie Yuan, Hongbo Fu, Yu-Kun Lai, and Hao Zhang. Sdm-net: Deep generative network for structured deformable mesh. *ACM Transactions on Graphics (TOG)*, 38(6):1–15, 2019. 4

[15] Aleksey Golovinskiy and Thomas Funkhouser. Consistent segmentation of 3d models. *Computers & Graphics*, 33(3):262–269, 2009. 4

[16] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6325–6334, 2017. 1, 3

[17] Songfang Han, Jiayuan Gu, Kaichun Mo, Li Yi, Siyu Hu, Xuejin Chen, and Hao Su. Compositionally generalizable 3d structure prediction. *arXiv preprint arXiv:2012.02493*, 2020. 4

[18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask r-cnn. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. 8

[19] Geoffrey Hinton. How to represent part-whole hierarchies in a neural network. *arXiv preprint arXiv:2102.12627*, 2021. 1, 10

[20] Yining Hong, Qing Li, Daniel Ciao, Siyuan Huang, and Song-Chun. Zhu. Learning by fixing: Solving math word problems with weak supervision. In *Thirty-Fifth AAAI Conference on Artificial Intelligence*, 2021. 3

[21] Yining Hong, Qing Li, Ran Gong, Daniel Ciao, Siyuan Huang, and Song-Chun. Zhu. Smart: A situation model for algebra story problems via attributed grammar. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI-21*, 2021. 3

[22] Yining Hong, Qing Li, Song-Chun Zhu, and Siyuan Huang. Vlgrammar: Grounded grammar induction of vision and language. *ICCV*, 2021. 4

[23] Ronghang Hu, Anna Rohrbach, Trevor Darrell, and Kate Saenko. Language-conditioned graph networks for relational reasoning. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10293–10302, 2019. 3, 7

[24] Ruizhen Hu, Oliver van Kaick, Bojian Wu, Hui Huang, Ariel Shamir, and Hao Zhang. Learning how objects function via co-analysis of interactions. *ACM Transactions on Graphics (TOG)*, 35 (4):1–13, 2016. 4

[25] Ruizhen Hu, Wenchao Li, Oliver Van Kaick, Ariel Shamir, Hao Zhang, and Hui Huang. Learning to predict part mobility from a single static snapshot. *ACM Transactions on Graphics (TOG)*, 36(6):1–13, 2017. 4

[26] Qixing Huang, Vladlen Koltun, and Leonidas Guibas. Joint shape segmentation with linear programming. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, pages 1–12, 2011. 4

[27] D. A. Hudson and Christopher D. Manning. Compositional attention networks for machine reasoning. *ICLR*, 2018. 3, 7

[28] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, 2019. 3

[29] J. Johnson, Bharath Hariharan, L. V. D. Maaten, Li Fei-Fei, C. L. Zitnick, and Ross B. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1988–1997, 2017. 1, 3

[30] Evangelos Kalogerakis, Aaron Hertzmann, and Karan Singh. Learning 3d mesh segmentation and labeling. In *ACM SIGGRAPH 2010 papers*, pages 1–12. 2010. 4

[31] Evangelos Kalogerakis, Siddhartha Chaudhuri, Daphne Koller, and Vladlen Koltun. A probabilistic model for component-based shape synthesis. *ACM Transactions on Graphics (TOG)*, 31 (4):1–11, 2012. 4

[32] Evangelos Kalogerakis, Melinos Averkiou, Subhransu Maji, and Siddhartha Chaudhuri. 3d shape segmentation with projective convolutional networks. In *proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3779–3788, 2017. 4

[33] Aishwarya Kamath, Mannat Singh, Y. LeCun, Ishan Misra, Gabriel Synnaeve, and Nicolas Carion. Mdetr - modulated detection for end-to-end multi-modal understanding. *ArXiv*, abs/2104.12763, 2021. 3, 7

[34] Dov Katz and Oliver Brock. Manipulating articulated objects with interactive perception. In *2008 IEEE International Conference on Robotics and Automation*, pages 272–277. IEEE, 2008. 4

[35] Vladimir G Kim, Wilmot Li, Niloy J Mitra, Siddhartha Chaudhuri, Stephen DiVerdi, and Thomas Funkhouser. Learning part-based templates from large collections of 3d shapes. *ACM Transactions on Graphics (TOG)*, 32(4):1–12, 2013. 4

[36] R. Krishna, Yuke Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, Stephanie Chen, Yannis Kalantidis, L. Li, D. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123:32–73, 2016. 3

[37] Jun Li, Kai Xu, Siddhartha Chaudhuri, Ersin Yumer, Hao Zhang, and Leonidas Guibas. Grass: Generative recursive autoencoders for shape structures. *ACM Transactions on Graphics (TOG)*, 36(4):1–14, 2017. 4

[38] Qing Li, Siyuan Huang, Yining Hong, Yixin Chen, Ying Nian Wu, and Song-Chun. Zhu. Closed loop neural-symbolic learning via integrating neural perception, grammar parsing, and symbolic reasoning. In *International Conference on Machine Learning (ICML)*, 2020. 3

[39] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention, 2020. 7, 8, 19

[40] Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. Inter-gps: Interpretable geometry problem solving with formal language and symbolic reasoning. In *The 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021. 3

[41] Tiange Luo, Kaichun Mo, Zhiao Huang, Jiarui Xu, Siyu Hu, Liwei Wang, and Hao Su. Learning to group: a bottom-up framework for 3d part discovery in unseen categories. *arXiv preprint arXiv:2002.06478*, 2020. 4

[42] Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B. Tenenbaum, and Jiajun Wu. The Neuro-Symbolic Concept Learner: Interpreting Scenes, Words, and Sentences From Natural Supervision. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJgMlhRctm. 3

[43] Roberto Martín-Martín, Clemens Eppner, and Oliver Brock. The rbo dataset of articulated objects and interactions. *The International Journal of Robotics Research*, 38(9):1013–1019, 2019. 4

[44] Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy Mitra, and Leonidas J Guibas. Structurenet: Hierarchical graph networks for 3d shape generation. *Siggraph Asia*, 2019. 4

[45] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 909–918, 2019. 2, 4

[46] Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy J Mitra, and Leonidas J Guibas. Structedit: Learning structural shape variations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8859–8868, 2020. 4

[47] O. Reiser and K. Koffka. Principles of gestalt psychology. *The Philosophical Review*, 45:412, 1936. 1

[48] Minjoon Seo, Hannaneh Hajishirzi, Ali Farhadi, and Oren Etzioni. Diagram understanding in geometry questions. In *AAAI*, 2014. 3

[49] Alane Suhr, Stephanie Zhou, Iris Zhang, Huajun Bai, and Yoav Artzi. A corpus for reasoning about natural language grounded in photographs. *ACL*, 2019. 1

[50] Minhyuk Sung, Hao Su, Vladimir G Kim, Siddhartha Chaudhuri, and Leonidas Guibas. Complementme: Weakly-supervised component suggestions for 3d modeling. *ACM Transactions on Graphics (TOG)*, 36(6):1–12, 2017. 4

[51] Ruocheng Wang, Jiayuan Mao, Samuel J. Gershman, and Jiajun Wu. Language-mediated, object-centric representation learning. *ACL (Findings)*, 2021. 4

[52] Xiaogang Wang, Bin Zhou, Yahao Shi, Xiaowu Chen, Qinping Zhao, and Kai Xu. Shape2motion: Joint analysis of motion parts and attributes from 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8876–8884, 2019. 4

[53] Zhijie Wu, Xiang Wang, Di Lin, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Sagnet: Structure-aware generative network for 3d-shape modeling. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019. 4

[54] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, et al. Sapien: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11097–11107, 2020. 4

[55] Zhige Xie, Kai Xu, Ligang Liu, and Yueshan Xiong. 3d shape segmentation and labeling via extreme learning machine. In *Computer graphics forum*, volume 33, pages 85–95. Wiley Online Library, 2014. 4

[56] Kexin Yi, Jiajun Wu, Chuang Gan, A. Torralba, P. Kohli, and J. Tenenbaum. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. In *NeurIPS*, 2018. 3, 7, 16

[57] Kexin Yi, Chuang Gan, Yunzhu Li, P. Kohli, Jiajun Wu, A. Torralba, and J. Tenenbaum. Clevrer: Collision events for video representation and reasoning. *ICLR*, 2020. 1

[58] Li Yi, Vladimir G Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (ToG)*, 35(6):1–12, 2016. 4

[59] Li Yi, Leonidas Guibas, Aaron Hertzmann, Vladimir G Kim, Hao Su, and Ersin Yumer. Learning hierarchical shape segmentation and labeling from online repositories. *SIGGRAPH*, 2017. 4

[60] Mehmet Ersin Yumer, Siddhartha Chaudhuri, Jessica K Hodgins, and Levent Burak Kara. Semantic shape editing using deformation handles. *ACM Transactions on Graphics (TOG)*, 34 (4):1–12, 2015. 4

[61] Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. From recognition to cognition: Visual commonsense reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6720–6731, 2019. 3

[62] Chi Zhang, Feng Gao, Baoxiong Jia, Yixin Zhu, and Song-Chun Zhu. Raven: A dataset for relational and analogical visual reasoning. In *CVPR*, pages 5317–5327, 2019. 3

[63] D. Zhang, Lei Wang, Nuo Xu, B. Dai, and H. Shen. The gap of semantic parsing: A survey on automatic math word problem solvers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42:2287–2305, 2020. 3

[64] Yuke Zhu, O. Groth, Michael S. Bernstein, and Li Fei-Fei. Visual7w: Grounded question answering in images. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4995–5004, 2016. 1, 3

A Basic Functions

Each question in PTR is associated with a functional program built from a set of basic functions. We detail the semantics of these functions.

A.1 Data Types

Our basic functional building blocks operate on values of the following types:

- `Object`: A single object in the scene.
- `ObjectSet`: A set of zero or more objects in the scene.
- `PartSet`: A set of parts in the scene.
- `Integer`: An integer between 0 and 9 (inclusive).
- `IntegerSet`: A set of integers.
- `Boolean`: `Yes` or `No`.
- `Value Types`:
  - `Object Category`: `Chair`, `Bed`, `Table`, `Refrigerator`, `Cart`
  - `Part Category`: `arm`, `leg`, `back`, `seat`, `central support pedestal`, `leg bar`, `wheel`, `arm vertical bar`, `arm horizontal bar`, `door`, `sleep area`, `top` , `drawer`, `shelf`, `body`
  - `Color`: `gray`, `red`, `blue`, `green`, `brown`, `purple`, `cyan`, `yellow`
  - `Stability`: `Stable`, `Unstable`
  - `Possible Change`: `to_left`, `to_right`, `to_front`, `to_behind`
- `Spatial Relationship`: `left`, `right`, `in front of`, `behind`, `above`, `below`
- `Geometric Relationship`: `line-line perpendicular`, `line-line parallel`, `plane-plane perpendicular`, `plane-plane parallel`, `line-plane perpendicular`, `line-plane parallel`

A.2 Object-Level Functions

Object-level functions focus on object-level reasoning, and are listed in Table 3.

A.3 Part-Level Functions

Since concepts, attributes and relationships are defined on the semantic level rather than instance level, we do not use a single `Part`. Rather, we use `PartSet` to denote both a set of parts of the same semantics, as well as a set of parts of different semantics. A dictionary keeps the correspondence between objects and parts (*e.g.*, {obj0: [part0, part1, part2], obj1: [part3, part4]...}), which facilitates hierarchical reasoning. Part-level functions are listed in Table 4. There are also arithmetic functions which focus on arithmetic problems on the number of parts, which are listed on Table 5.

Function	Description	Input	Output
scene	Returns the set of all objects in the scene	∅	ObjectSet
unique	If the input is a singleton set, then return it as an Object	ObjectSet	Object
relate_object	Return all objects in the scene that have the specified spatial relation to the input object	Object, Spatial Relationship	ObjectSet
count_object	Returns the size of the input set	ObjectSet	Integer
exist_object	Returns yes If the input set is nonempty and no if it is empty	ObjectSet	Boolean
filter_object_category	Filter objects by category	ObjectSet, Object Category	ObjectSet
filter_stability	Filter objects by stability	ObjectSet, Stability	ObjectSet
query_object_category	Query the category of an object	Object	Object Category
query_possible_change	Query possible changes for the objects to stay stable	Object	Possible Change
query_is_stability	Query whether an object is stable or unstable	Object, Stability	Boolean
query_is_possible_change	Query whether applying possible change to an object can make it stable	Object, Possible Change	Boolean
same_object_category	Return the set of objects that are of the same category as the inpuy	Object	ObjectSet
equal_object_category	Check whether two categories are the same	Object Category, Object Category	Boolean
query_object_analogy	The first Object A and the second Object B have certain spatial relationships. We want to find the fourth Object D so that the third Object C and the fourth Object D have the same	Object, Object, Object	Object

Table 3: Object-Level Functions

B Human Performance

To evaluate human performances for our dataset, we hire graduate students in a university, at a rate of $15/hr. We first provide the subjects with detailed concepts from PTR dataset, and sample questions and answers for the subjects to get familiar with the dataset. We present 2500 random questions from the test set, and take a majority vote among three subjects for each question.

C NS-VQA details

For NS-VQA, we first use Mask-RCNN to propose segmentations for objects and parts. Next, we extract attributes using attribute networks, which have the same parameters as in [56]. Object segmentations are sent to the attribute net to predict the categories of the objects. To facilitate reasoning on questions about spatial relationships, we follow [56], augment the objects with the original images, and use the attribute net to predict the coordinates of each object. Object proposals augmented with original images are also used to predict the stability (stable, unstable) of each object. If an object is unstable, possible changes (to_left, to_right, to_front, to_behind) are predicted.

And then, the part proposals are augmented with the predicted object proposals and attributes, to predict the part attributes. Specifically, if the part is within the region of an object proposal, we limit the categories of the parts to the specific part categories of the object categories (*e.g.*,, if the predicted object is cart, we limit the part categories to body and wheel). The part proposals are also augmented with the object proposals and fed to the attribute networks. To predict geometric information, the

Function	Description	Input	Output
expand_parts	Return the parts of a set of objects.	ObjectSet	PartSet
relate_part	Return the set of parts that have the certain geometric relationship to the partset	PartSet, Geometric Relationship	PartSet
exist_part	Return Yes if the PartSet is not empty	PartSet	Boolean
filter_part_category	Filter parts by category	PartSet	PartSet
filter_part_color	Filter parts by color	PartSet	PartSet
count_part	Count the number of certain parts in each object. Specifically, using the object-part dictionary, count how many parts of each object are in the PartSet, return as IntegerSet. IntegerSet has the same size as the ObjectSet returned by **scene**	PartSet	IntegerSet
filter_part_exist	Filter the set of objects that contain parts in the PartSet	PartSet	ObjectSet
filter_part_count	Filter a set of objects that have a number of parts that matches the input Integer. Specifically, **count_part** returns the number of parts in the PARTSET of each object as IntegerSet. **filter_part_count** takes IntegerSet and filters objects by checking whether the corresponding number of parts in the IntegerSet matches the input Integer.	IntegerSet	ObjectSet
query_part_category	query the category of parts	PartSet	Part Category
query_part_color	query the color of parts	PartSet	Color
same_part_color	Return the set of parts with the same color as input	PartSet	PartSet
equal_part_category	Check whether two categories are the same	Part Category, Part Category	Boolean
equal_part_color	Check whether two colors are the same	Color, Color	Boolean
query_part_analogy	The first PartSet A and the second PartSet B have certain spatial relationships. We want to find the fourth PartSet D so that the third PartSet C and the fourth PartSet D have the same relationships.	PartSet, PartSet, PartSet	PartSet

Table 4: Part-Level Functions

Function	Description	Input	Output
equal_integer	Check whether two integers are equal	`Integer`, `Integer`	`Boolean`
greater_than	Check whether the first integer is greater than the second	`Integer`, `Integer`	`Boolean`
less_than	Check whether the first integer is smaller than the second	`Integer`, `Integer`	`Boolean`
sum	Return the sum of two integers	`Integer`, `Integer`	`Integer`
minus	Return the difference between two integers	`Integer`, `Integer`	`Integer`

Table 5: Arithmetic Functions

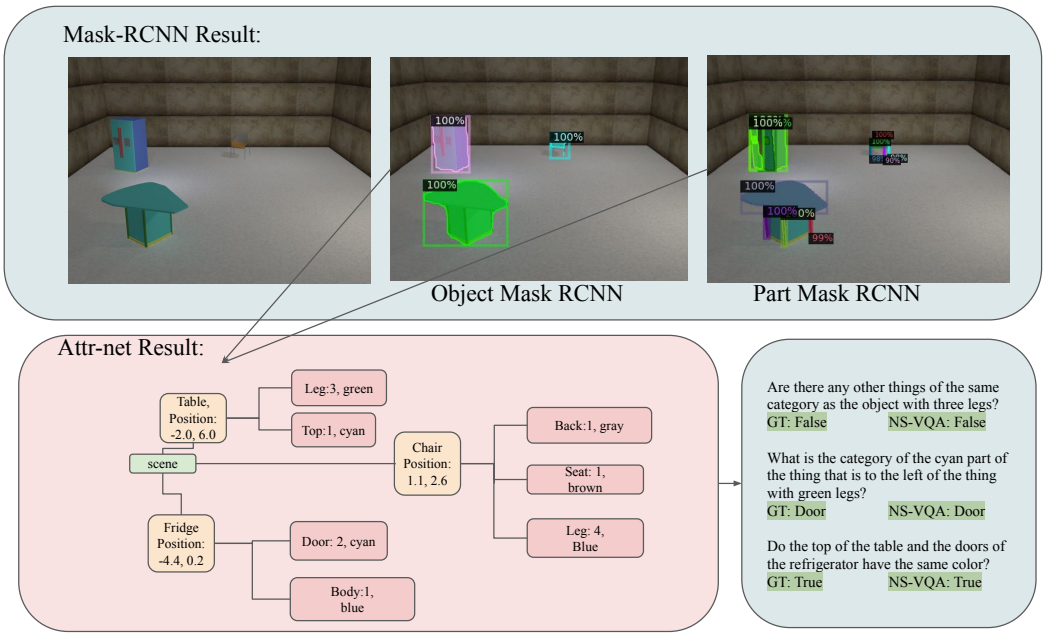


Figure 7: An example of NS-VQA.

attribute network first outputs probabilities suggesting whether the parts can be considered lines or planes. If true, the network outputs a three-dimensional vector, which denotes the parameters of the line/plane equation. If a part proposal is not within the region of an object proposal, we first check whether it's connected to an object proposal, or that the distance between the part proposal and the object proposal is smaller than a threshold. If so, we add the region of the part proposal to the object proposal, treat the part as a part within the region of an object proposal, and predict the attributes of the parts as above. If the part is not within, connected to or close to an object proposal, we first augment the part proposal with a bounding box that is greater than the part proposal and covers the part proposal, to represent the contextual information. The part proposal is then sent to the attribute network to predict its attributes. The part categories are not limited in this case. Then, we group these isolated part proposals if they are connected to or close to each other in a threshold. We take the output probabilities of part categories. We calculate the probabilities of object categories by summing up the maximum probabilities of part categories that belong to the corresponding object categories. We determine the categories of the objects using these probabilities, and then determine the part categories according the object categories and the probabilities of part categories.

In Figure 7, we provide an example of the inputs and outputs of NS-VQA.

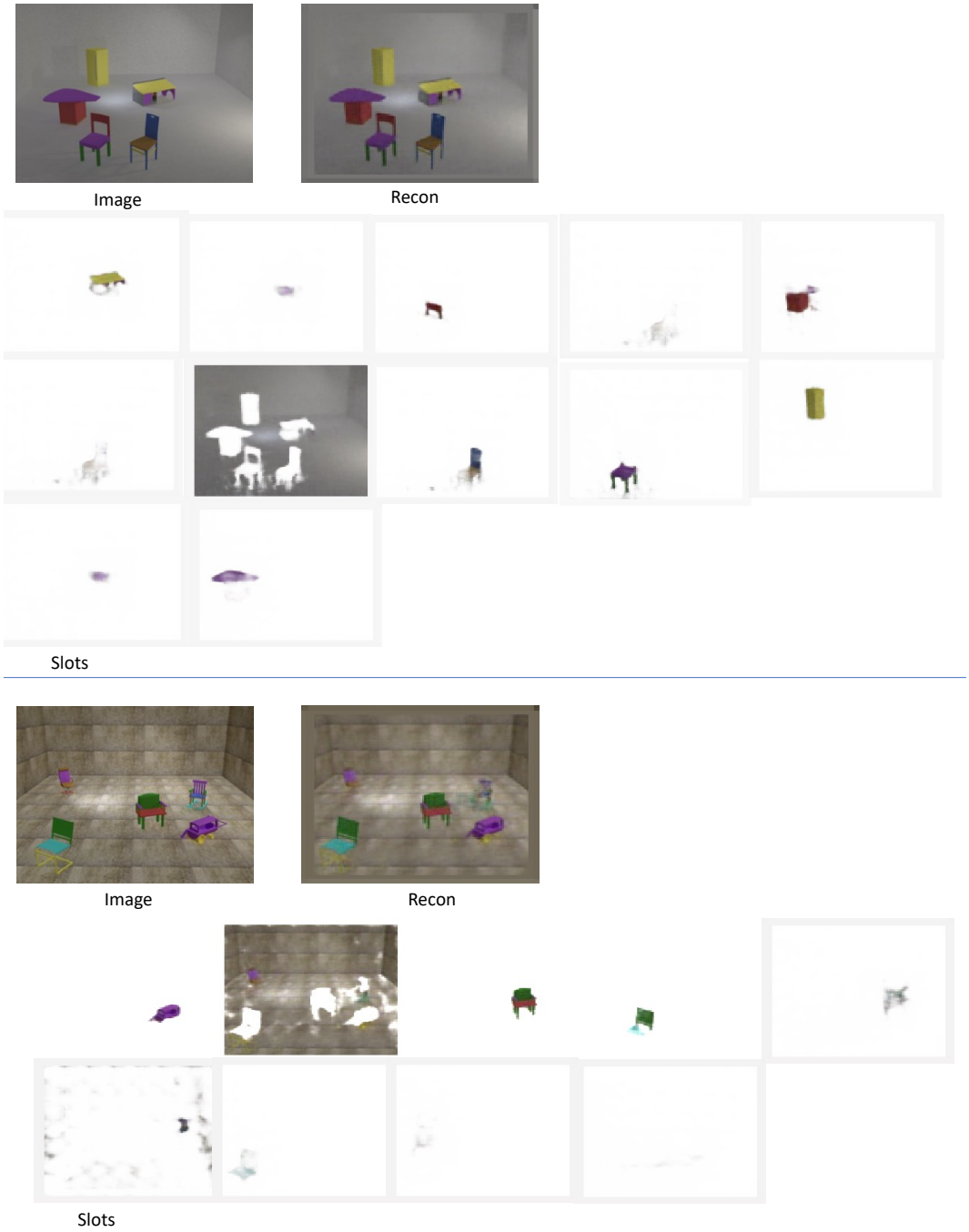


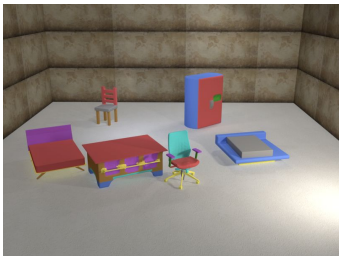
Figure 8: Slot Attention Results

D Unsupervised Part-Centric Rrepresentations

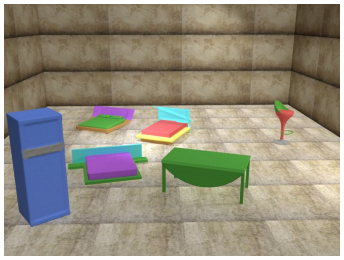
Here we show unsupervised segmentation results provided by Slot Attention[39], which is shown in Figure 8.

Concept

Q: *How many things with wheels are there?*
A: *2*
Q-type: *count_object*



Q: *How many legs does the object with one central support have?*
A: *5*
Q-type: *count_part*

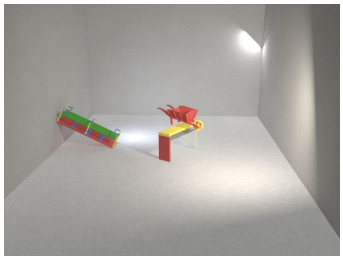


Q: *How many beds with cyan back are there?*
A: *2*
Q-type: *count_object*

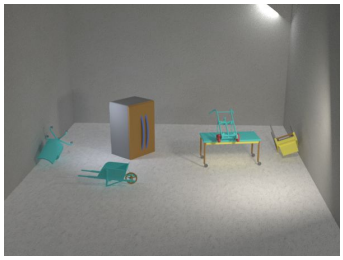
Q: *Are there any objects with two cyan legs?*
A: *Yes*
Q-type: *exist_object*

Q: *What is the color of the sleep area of the bed with brown legs?*
A: *Red*
Q-type: *query_part*

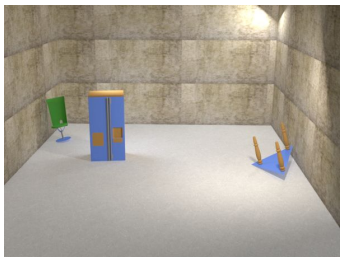
Q: *What is the category of the thing with legs?*
A: *Table*
Q-type: *query_object*



Q: *How many tables with shelves are there?*
A: *2*
Q-type: *count_object*



Q: *What is the color of the top of the table with brown legs?*
A: *cyan*
Q-type: *query_part*



Q: *What is the color of the pedestal of the chair?*
A: *Blue*
Q-type: *query_part*

Q: *Are there any objects with yellow legs?*
A: *Yes*
Q-type: *exist_object*

Q: *How many doors does the refrigerator have?*
A: *2*
Q-type: *count_part*

Q: *How many legs does the table have?*
A: *3*
Q-type: *count_part*

Figure 9: Exemplar images and questions on conceptual problems

E Examples

Randomly chosen exemplar images, questions and answers are shown in Figure 9-12. We categorize the questions using the semantics types of the questions. We also show the sub-type of each question, denoted as Q-type in the figures.

Relation

Q: *What is the number of the visible legs of the thing that has the same color of back as the chair with one gray seat?*
A: *3*
Q-type: *same_relate*

Q: *What is the category of the object with brown legs that is on the right side of the chair with a pedestal?*
A: *Table*
Q-type: *spatial_relationship*

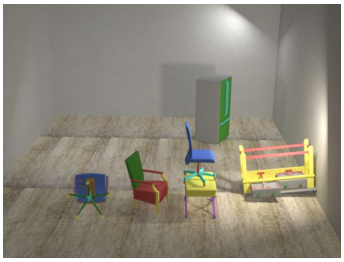
Q: *Is there a part in the chair that can be considered a line, and is parallel to the brown part of the refrigerator?*
A: *No*
Q-type: *geometric_relationship*

Q: *Are the object with purple legs and the object with gray legs of the same category?*
A: *No*
Q-type: *same_relate*

Q: *What is the category of the green part of the object with four visible legs that is on the left side of the chair with two arm vertical bars?*
A: *Leg bar*
Q-type: *spatial_relationship*

Q: *What is the color of the part in the chair that can be considered a plane, and is perpendicular to the brown part of the refrigerator?*
A: *Yellow*
Q-type: *geometric_relationship*

Q: *What is the color of the legs of the thing that has the same color of back as the bed?*
A: *Brown*
Q-type: *same_relate*



Q: *How many things with legs are in front of the refrigerator?*
A: *5*
Q-type: *spatial_relationship*

Q: *Are the body of the refrigerator, and the central support of the chair with pedestal of the same color?*
A: *Yes*
Q-type: *same_relate*

Q: *What is the color of the leg bars of the thing with yellow legs that is to the right of the table with yellow top?*
A: *Red*
Q-type: *spatial_relationship*

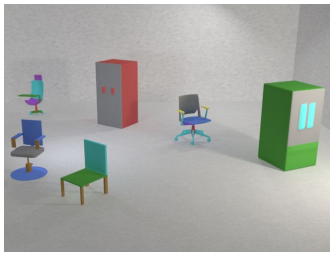
Figure 10: Exemplar images and questions on relational problems

Analogy

Q: *The thing with five legs has certain positional relation to the object with blue seat. By analogy, how many objects does the bed have the same positional relation to ?*
A: *2*
Q-type: *positional analogy*

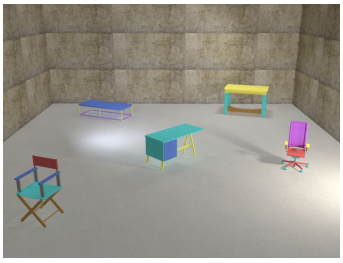
Q: *The cart has certain positional relation to the chair with central support. By analogy, is there an object that the chair with green seat has the same positional relation to?*
A: *Yes*
Q-type: *positional_analogy*

Q: *The purple part of the table with three gray legs has certain geometric relation to the red part of the table with one brown top. by analogy, the blue part of the refrigerator with cyan body has the same geometric to the part of which color in the refrigerator with purple body?*
A: *gray*
Q-type: *geometric_analogy*



Q: *The yellow parts of the chair with gray back have certain geometric relation to the gray part of the refrigerator with green body. By analogy, the brown parts of the chair with green seat have the same geometric relation to a part of what color in the chair with cyan legs?*
A: *Blue*
Q-type: *geometric_analogy*

Arithmetic



Q: *What is the sum of the legs in the table with blue top, and the number of drawers in the table with cyan top?*
A: *9*
Q-type: *sum-minus*

Q: *Are there an equal number of doors in the refrigerator, and leg bars in the chair with cyan seat?*
A: *Yes*
Q-type: *compare-integer*

Q: *Are there fewer legs in the table with purple top, than drawers in the table with cyan legs?*
A: *No*
Q-type: *compare-integer*

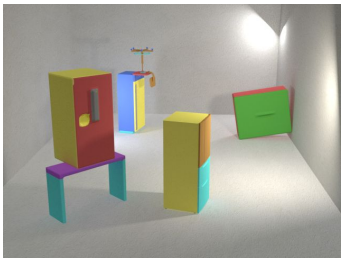
Q: *What is the number of arm vertical bars in the chair with brown legs, subtracted by the number of drawers in the table with cyan top?*
A: *1*
Q-type: *sum-minus*

Q: *Are there fewer legs in the chair with gray seat, than legs in the chair with yellow seat?*
A: *No*
Q-type: *compare-integer*

Q: *What is the number of drawers in the table with cyan legs, subtracted from the number of legs in the table with purple top?*
A: *2*
Q-type: *sum-minus*

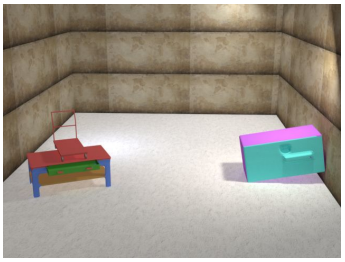
Figure 11: Exemplar images and questions on analogical and arithmetic problems

Physics



Q: *Is the chair stable?*
A: *No*
Q-type: *stability*

Q: *Towards which direction should the chair move to become stable?*
A: *To left*
Q-type: *possible changes*



Q: *How many objects are stable?*
A: 1
Q-type: *stability*

Q: *Can moving front makes the cart stable?*
A: *no*
Q-type: *possible changes*

Q:*Is the table with yellow top stable?*
A: *No*
Q-type: *stability*

Q: *Is the bed stable?*
A: *Yes*
Q-type: *stability*

Figure 12: Exemplar images and questions on physical problems