StructFormer: Learning Spatial Structure for Language-Guided Semantic Rearrangement of Novel Objects

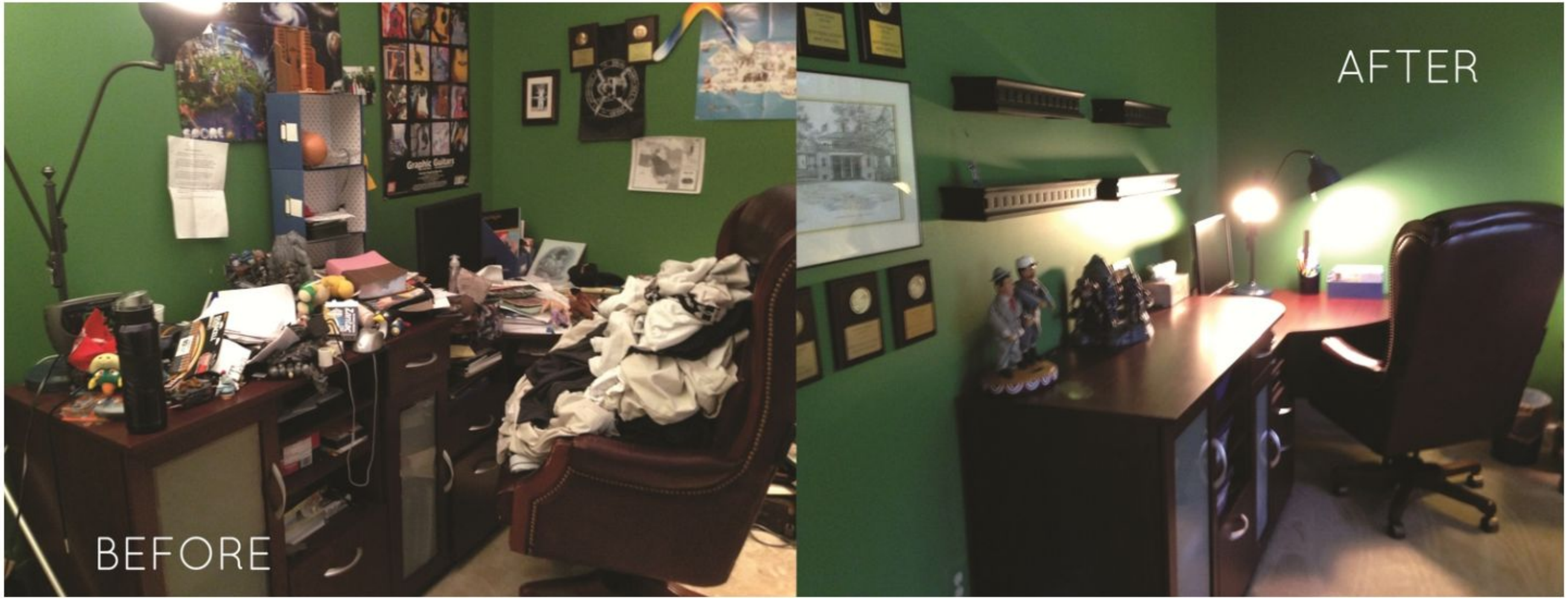
Weiyu Liu, Chris Paxton, Tucker Hermans and Dieter Fox

Presenter: Sharath

27 October 2022

Motivation

Can we have a robot do this for us?

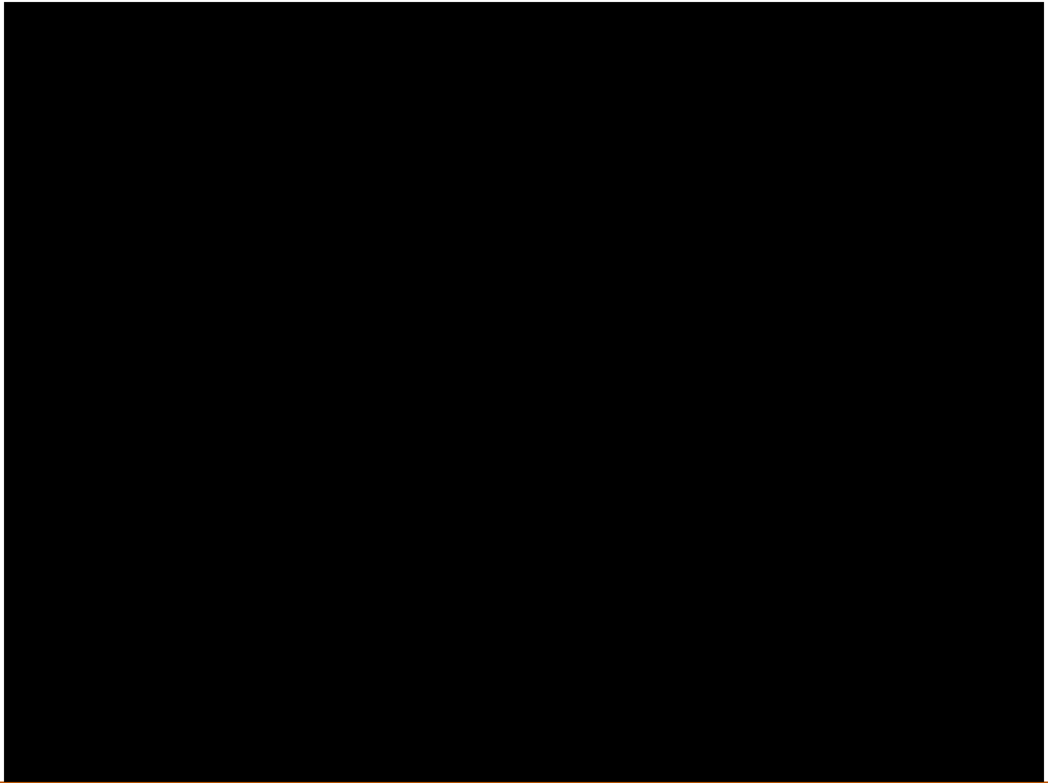


Motivation

Probably not…

What if it can do the following from just voice commands?

It's a great start!

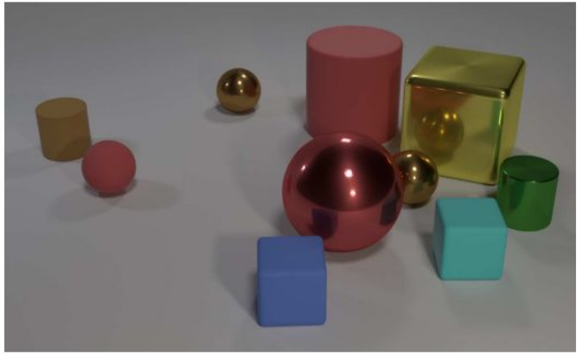


Problem Setting

- Geometric organization of objects into semantically meaningful arrangements pervades the built world. As such, assistive robots operating in warehouses, offices, and homes would greatly benefit from the **ability to recognize and rearrange objects into these semantically meaningful structures**.
- StructFormer, takes as input a **partial-view point cloud** of the current object arrangement and a **structured language command** encoding the desired object configuration to arrange objects into complex structures such as circles or table settings.

Prior Work

1. Visual reasoning systems. Passive, does not translate to control.
2. Rearrangement of pairwise objects. No joint reasoning of multiple objects.
3. Images goals dictate desired rearrangement. Not language guided.



Q: Are there an equal number of large things and metal spheres?

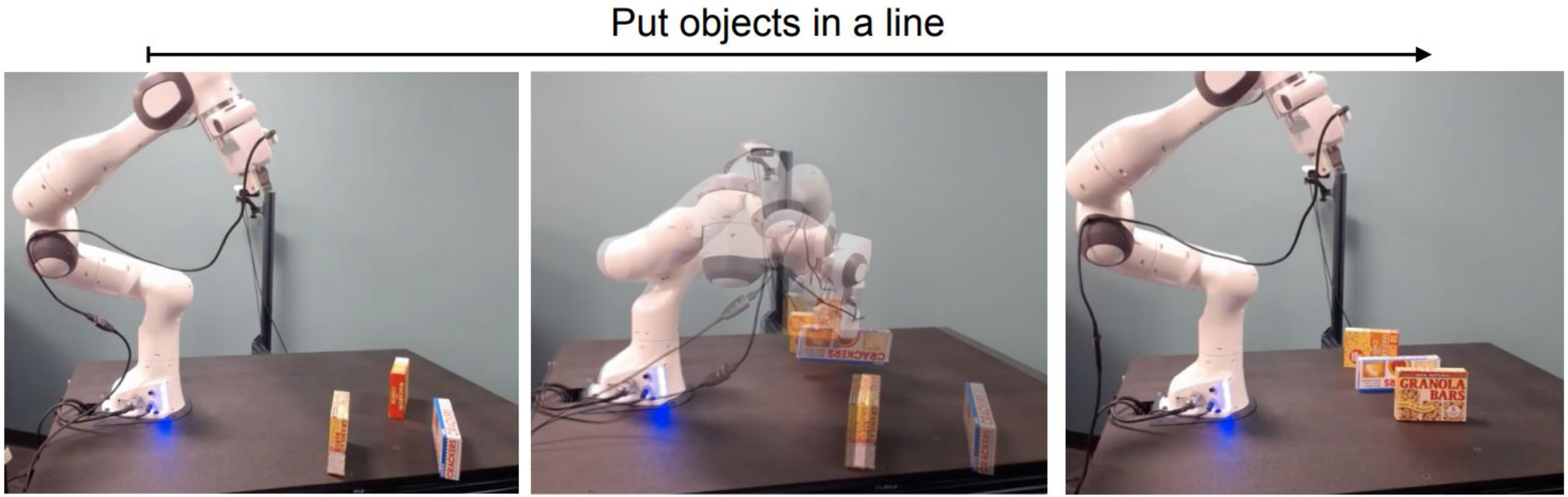
1. Johnson et al, 2017

2. Paxton et al, 2021

3. Qureshi et al, 2021

Proposed Work

Rearrange through manipulation unknown objects into semantically meaningful multi-object spatial structure as dictated by input language commands.



Proposed Approach

Semantic Rearrangement Problem

as

Sequential Prediction Task

using

Novel Transformer Architecture

Word Embeddings

Latent Rep	$\tilde{c}_i$	same	class	yellow	mug	circle	bottom	right	large
Pos Emb	p_i	1	2	3	4	5	6	7	8
Type Emb	r_i	0	0	0	0	0	0	0	0

Latent Representation

Learnt Word Embeddings

Tokenization

"Rearrange objects that have the same class as the yellow object into a large circle at the bottom right of the table"

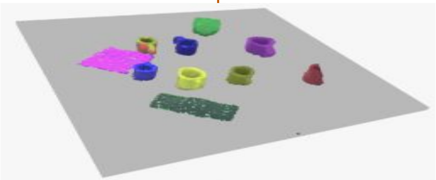
Language Instruction

Point Cloud Transformer

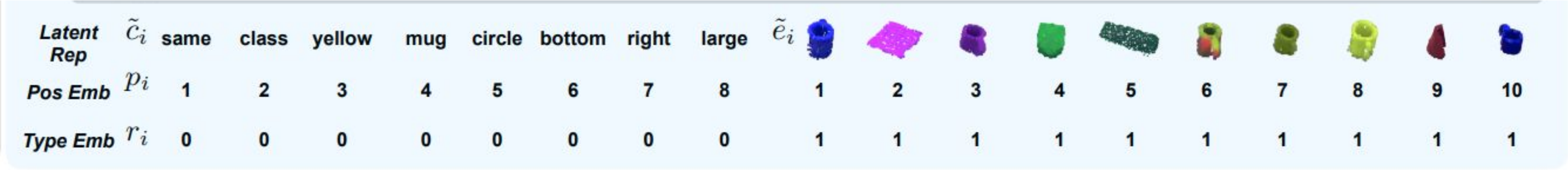
Latent Representation

Point Cloud Transformer (PCT)

Segmented Point Cloud



Position and Type embeddings

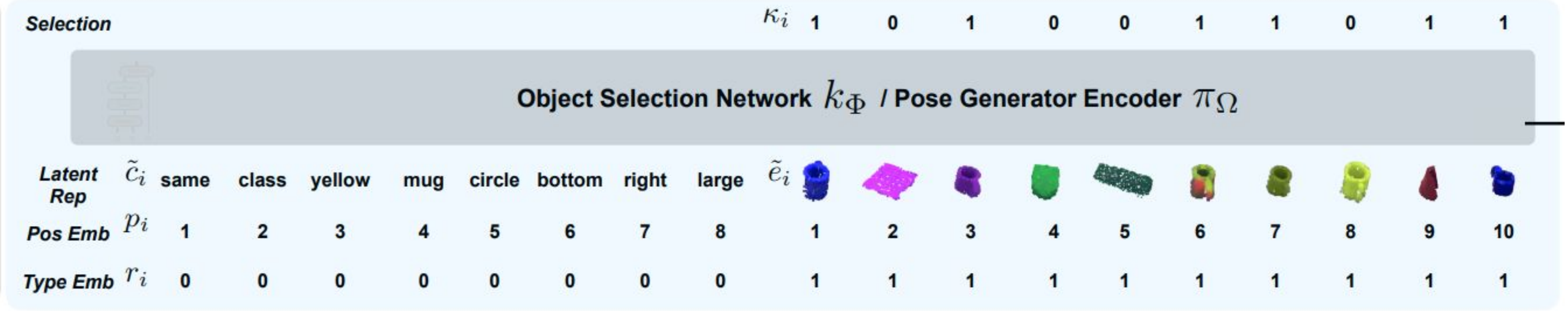


Position embedding: Position of word/object in sequence

Type embedding: Word or object?

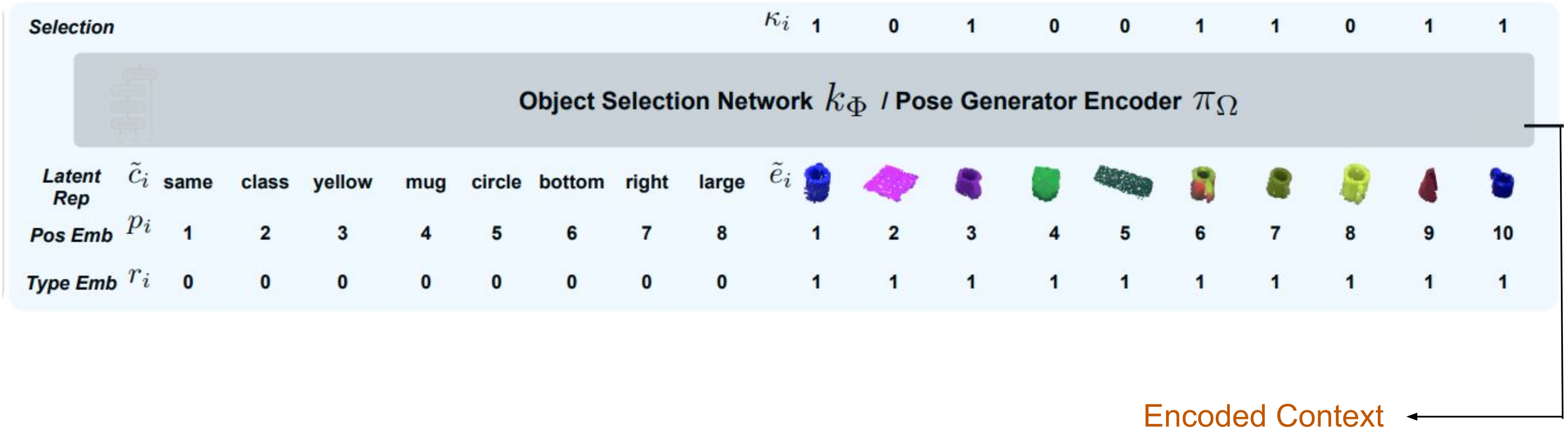
Object Selection Network

Output binary sequence: Should object be moved?



Input embeddings

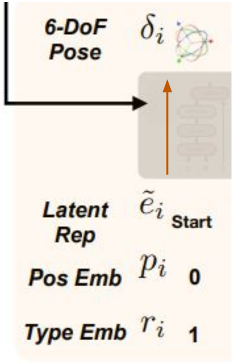
Pose Generator Encoder



Pose Generator Decoder

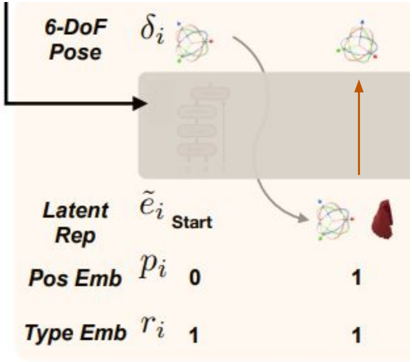
Predicted Pose Offset of structure frame

Initial Condition



Pose Generator Decoder

Predicted Pose Offset of previous object



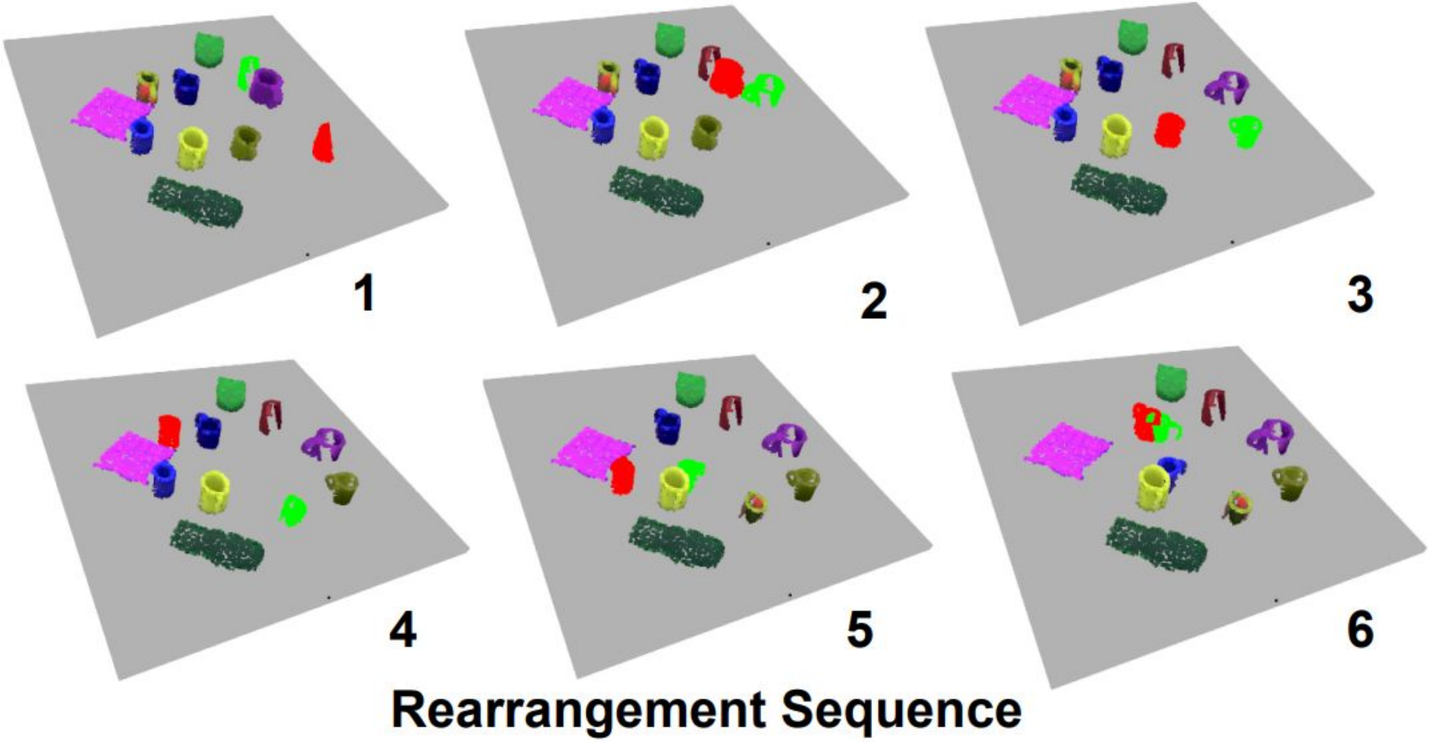
Predicted Pose Offset of “query” object

t: position offset w.r.t structure frame
R: rotation offset between target and initial pose

Object embedding of “query” object

Chosen by object selection network previously

Example output



Inference and Training

- Inference:
  - Objects are sampled with the object selection network
  - Autoregressively predict the target pose offset
- Training (supervised learning):
  - The object selection network is trained on initial scenes and ground truth query objects using a binary cross entropy loss.
  - The generator is trained with an L2-loss minimizing the distance between groundtruth (from rearrangement sequence data) and predicted placement poses

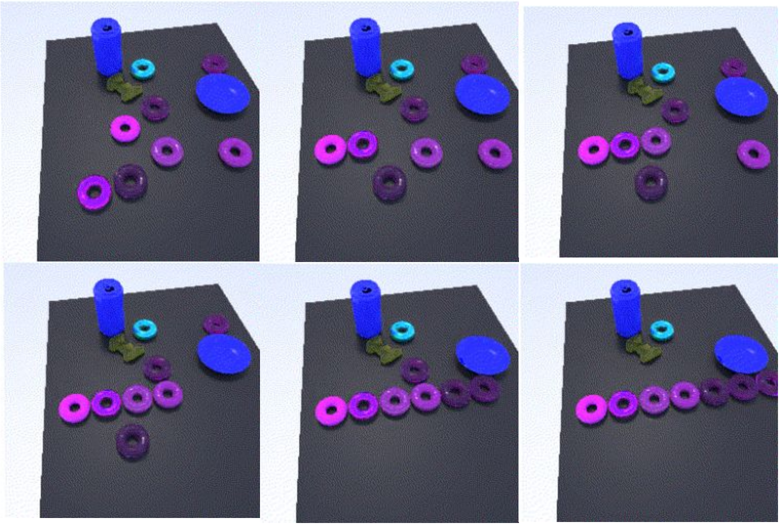
Data Generation

- 100,000 x {*segmented point cloud of rearrangement sequence, language instruction of target spatial rearrangement*}
- 35 object classes with a total of 323 objects
- PyBullet physics simulator for placing objects and NVISII for rendering color and depth images with instance segmentation masks for all objects.
- Language instructions created from the pool shown here.

Entity	Type (# Value)	Values
obj	class (35)	basket, beer bottle, book, bowl, calculator, candle, controller, cup, donut, ...
	material (3)	glass, metal, plastic
	color (6)	blue, cyan, green, magenta, red, yellow
	relate (3)	less, equal, more
struct	shape (4)	circle, line, tower, table setting
	size (3)	small, medium, large
	vertical position (3)	top, middle, bottom
	horizontal position (3)	left, center, right
	rotation (4)	north, east, south, west

Data Generation

- Example datapoint:



Rearrange objects that have the same size as the metal, cyan donut into medium line in the middle center of the table.

Data Generation

Procedure:

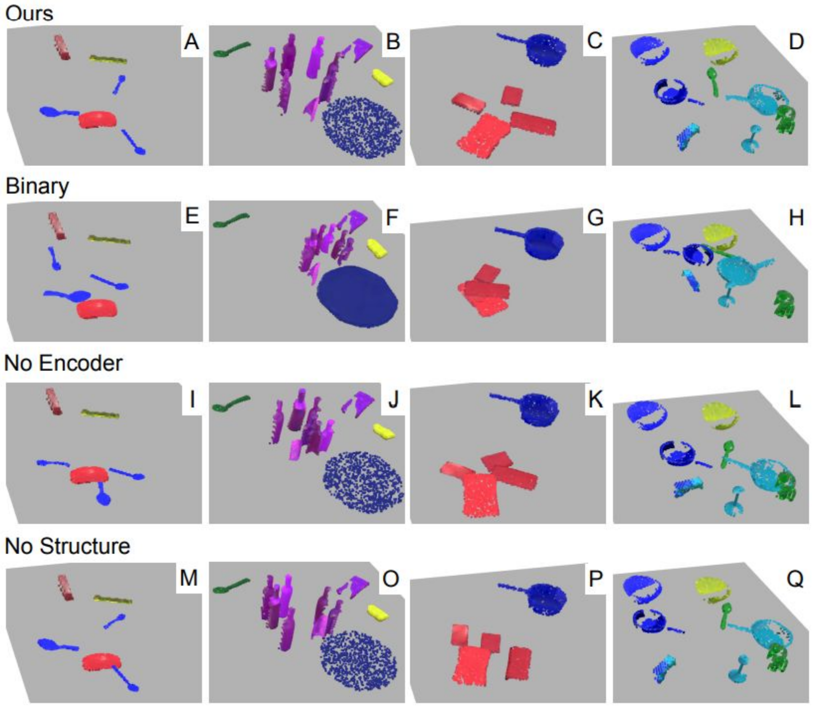
1. Sample a referring expression for query object
    - e.g., objects that have the same material as the blue bottle
2. Manually rearrange into one of the four predefined spatial structures
3. Generate a sentence using the selected structure and reference object
    - e.g., place query objects into a large circle on the top right of the table)
4. Move objects one-by-one randomly out of the scene. The reverse of this is the reference rearrangement sequence.

Baselines

1. Binary
   - An object is rearranged using the point cloud of the previously rearranged object. This is done iteratively.
   - Allows to evaluate the efficacy of modeling pairwise relations for our task.
2. No Encoder
   - An encoded global context is not used. Instead it relies on the language instruction.
3. No Structure
   - Pose-offset of the virtual structure frame is not predicted or used

Experimental Results - Comparative

- **No Encoder** and **Binary** baselines cannot produce circular structures as they lack global context.
- **No Struct** fails to place objects precisely

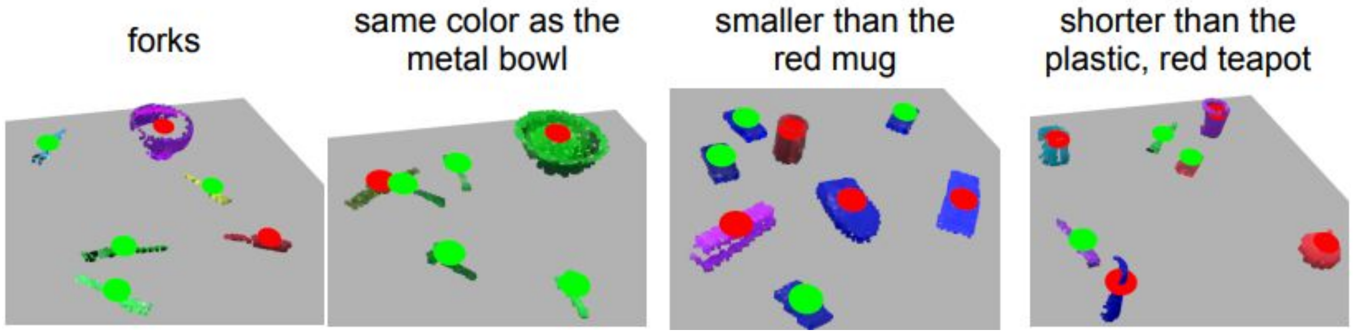


Experimental Results - Comparative

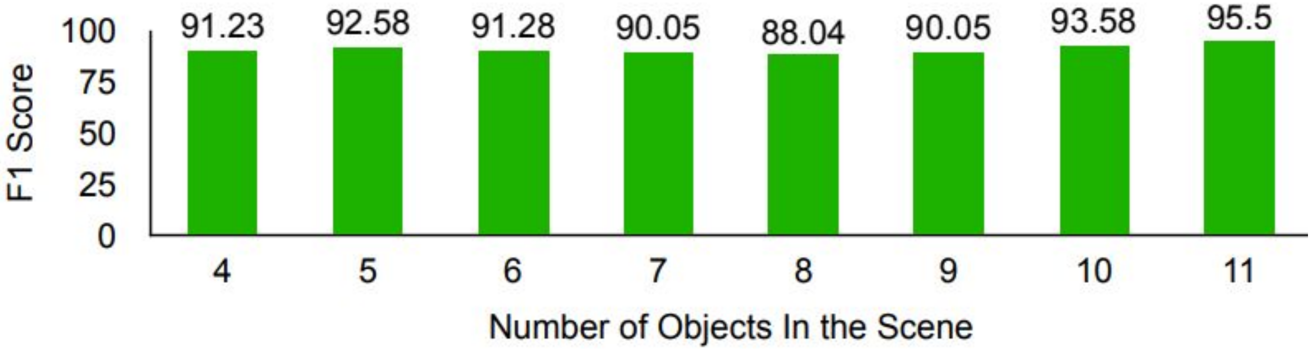
- Testing metrics:
  - Euclidean distance for position errors
  - Geodesic distance/ rotation about equivalent axis for orientation errors
- **StructFormer** outperforms baselines for all four spatial structures.

Experimental Results-Object Selection Performance

Qualitative

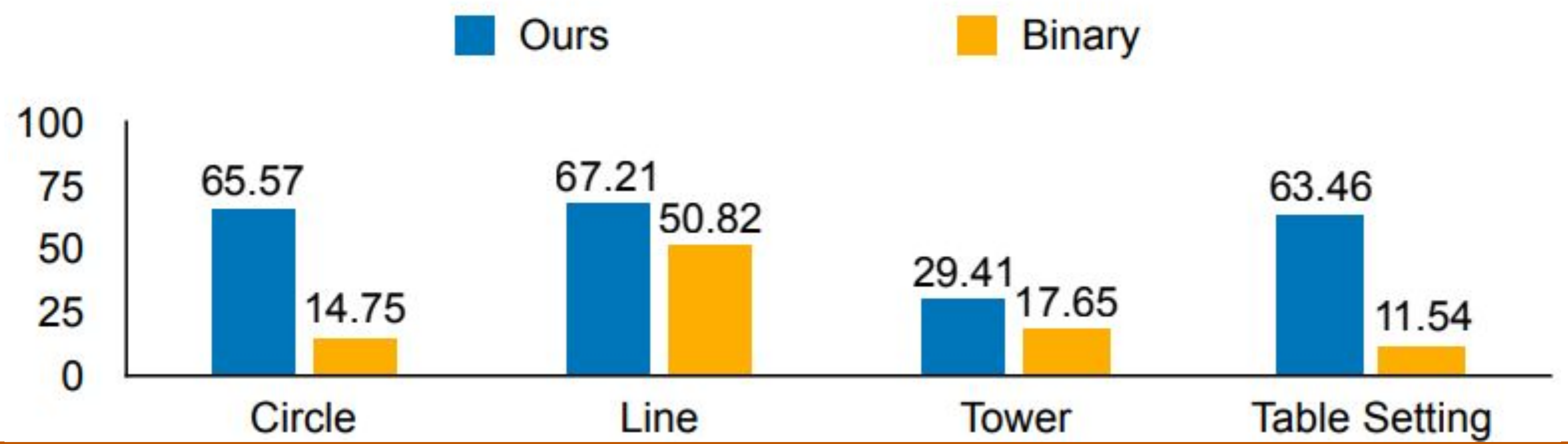


Quantitative



Experimental Results - Full system in Simulation

- Test data generated using 138 novel object models from 23 known object classes. E.g., red book not present in training set by blue book
- The overall success rate of the whole pipeline was 58/156 (37%).
- Comparison with the Binary algorithm is shown below



Discussion of Results

- Usefulness of estimating pose offset of structure: to deal with spatial ambiguities embedded in language instructions (e.g., arrange a circle in the middle of the table).
- The performance difference between Binary and StructFormer shows that modeling multi-object spatial relations is beneficial for creating complex spatial structures
- Works well with large number of objects in the scene

Critique

- Overall success rate in simulation was only 37%?!
  - What will the success rate be in the real world?
- No extensive testing results on the physical robot.
  - Paper promised capabilities in cluttered scenarios and for complex spatial structures, but real world demonstrations are shown for simple setups
  - Occlusions create issues for the robot
- Architecture is large, complex and uninterpretable.
  - Must have been difficult to train and tune hyperparameters
- Getting a robot to tidy my room is years away!

Future Work for Paper / Reading

- In the rearrangement sequence specified in the paper, a chosen object is moved only once.
    - A very cluttered scene would require object to be manipulated more than once
- Evaluation the performance with a complete perception-planning-control pipeline in the real world.
- The algorithm predefined the order of rearrangement rather that finding the optimal one
- Breaking down the algorithm into two parts:
    - Part one decides the sequence in which objects need to be moved
    - Part two computes the execution of each step in the sequence (in a hierarchical way)

Extended Readings

- Transformers are Adaptable Task Planners:
  - One-shot learning of new demonstrations
- DALL-E-Bot: Introducing Web-Scale Diffusion Models to Robotics:
  - Zero-shot rearrangement using DALL-E
- Code as Policies: Language Model Programs for Embodied Control:
  - Language models are used to generate policy code

Summary

- Paper addresses the problem of actively rearranging unknown objects into semantically meaningful multi-object spatial structures based on high-level language instructions.
- This is a difficult problem because it involves complex spatial reasoning
- Previous work could only perceive and not act
- The proposed work jointly reasons about multiple objects that results in better predictions
- They show with both simulations and real world experiments that their proposed architecture is sufficient for task.