

# ConSOR: A Context-Aware Semantic Object Rearrangement Framework for Partially Arranged Scenes

Kartik Ramachandruni, Max Zuo, and Sonia Chernova

**Abstract**—Object rearrangement is the problem of enabling a robot to identify the correct object placement in a complex environment. Prior work on object rearrangement has explored a diverse set of techniques for following user instructions to achieve some desired goal state. Logical predicates, images of the goal scene, and natural language descriptions have all been used to instruct a robot in how to arrange objects. In this work, we argue that burdening the user with specifying goal scenes is not necessary in partially-arranged environments, such as common household settings. Instead, we show that contextual cues from partially arranged scenes (i.e., the placement of some number of pre-arranged objects in the environment) provide sufficient context to enable robots to perform object rearrangement *without any explicit user goal specification*. We introduce ConSOR, a Context-aware Semantic Object Rearrangement framework that utilizes contextual cues from a partially arranged initial state of the environment to complete the arrangement of new objects, without explicit goal specification from the user. We demonstrate that ConSOR strongly outperforms two baselines in generalizing to novel object arrangements and unseen object categories. The code and data are available at <https://github.com/kartikvrama/conson>.

## I. INTRODUCTION

Consider a service robot tasked with putting away newly delivered groceries, or cleaning a living room. In both tasks, the environment is most likely already partially arranged, and that arrangement provides valuable clues for where new items should be placed. For example, the pantry may already contain unfinished boxes of cereals and pasta on different shelves, while the left drawer of the refrigerator may contain half-finished vegetables. Thus, new items, such as a box of oatmeal, should be placed in accordance with the user’s existing organization scheme (e.g., near the cereal). Similarly, a book may naturally be placed alongside other books on the shelf rather than next to houseplants.

The general problem of identifying the correct item placement in a complex environment is known as the *object rearrangement problem* [1]. Prior work on object rearrangement has explored a diverse set of techniques for following user instructions to achieve some desired goal state. Logical predicates [2], [3], images of the goal scene [4], [5], [6], and natural language descriptions [7], [8], [9] have all been used to instruct a robot in how to arrange objects. However, all of the above techniques place a burden on the user to explicitly describe the goal state, or else to explicitly demonstrate the rearrangement task so that the robot can learn from demonstrations [10], [11].

Georgia Institute of Technology, Atlanta, Georgia, United States. Contact: {kvr6, zuo, chernova}@gatech.edu

In this work, we posit that contextual cues from partially arranged scenes (i.e., the placement of some number of pre-arranged objects in the environment) provide sufficient context to enable robots to perform object rearrangement *without any explicit user goal specification*. Closely related to our work are those of Abdo et al. [12] and Wu et al. [13], which reason about object similarities by learning object relationships from demonstrations of arranged environments, which are then generalized to novel environments. However, these works require that the desired organizational style in the goal state be known *a priori* (e.g., specified by the user) instead of inferring this style from scene context.

We introduce ConSOR, a Context-aware Semantic Object Rearrangement framework that utilizes contextual cues from a partially arranged initial state of the environment to complete the arrangement of new objects, without explicit goal specification from the user. Figure 1 presents an overview of our framework. ConSOR reasons about the semantic properties of objects in the environment, and the context provided by the number of containers and existing placement of objects into containers, to infer the desired placement for new, unarranged objects. Additionally, ConSOR leverages prior commonsense knowledge from pre-trained ConceptNet embeddings to perform zero-shot generalization to scenes with objects unseen during training. Our work makes the following contributions:

- We formalize the problem of object rearrangement in partially arranged environments.
- We present ConSOR, a Context-aware Semantic Object Rearrangement framework that replaces human instruction with contextual cues from the initial state of the environment to infer the desired goal state of an object rearrangement task.
- We contribute a dataset of 8k rearranged goal states from a dataset of 38 household objects, with each goal state associated with one of four predefined organizational *schemas*.
- We demonstrate that ConSOR is able to generalize both to novel arrangements and novel object classes, achieving high performance across all four organizational schemas we tested.

We compare ConSOR with two baselines, a collaborative filtering-based approach to grouping objects based on learned pairwise similarity scores [12] and the GPT-3 large language model [14], on a withheld set of novel object arrangements and object types. Our results show that ConSOR strongly outperforms both baselines in every tested category, without

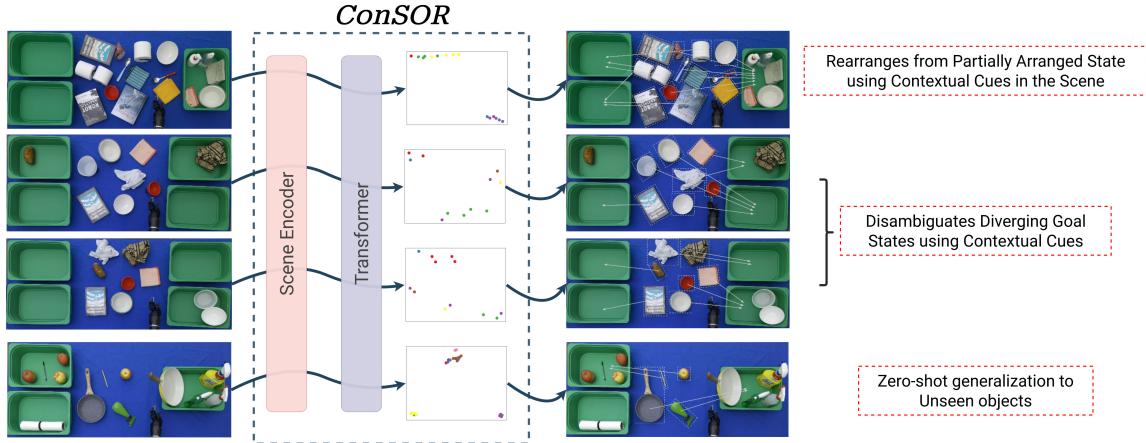


Fig. 1. Our semantic object rearrangement framework, ConSOR, takes partially arranged object scenes (left), uses a Transformer-based neural architecture to infer contextual cues about the likely goal arrangement, and generates the desired arrangement state (right).

assuming that the target organization scheme is known *a priori*.

## II. RELATED WORK

Numerous works in the literature have proposed approaches to goal-conditioned object rearrangement. The means by which a user specifies the goal varies. In some works, the goal is represented by a set of logical predicates encoding relationships between objects [2], [3]. Alternately, in *visual object rearrangement*, the goal is specified as an image, and the robot must perform object matching between the initial and goal images to determine the required object placement [4], [5], [6]. A third form of goal specification is *natural language instruction*, and recent work in language-conditioned manipulation has contributed techniques that ground a language description of the desired goal to the observable environment while performing zero-shot generalization to novel language commands [7], [8], [9], [15]. Critically, all of the above methods are ineffective in the absence of an explicit goal specification.

To perform rearrangement without goal specification, some recent works take the approach of learning user-specific preferences, often modeling these preferences from a single demonstration [10], [11]. These methods translate preferences encoded in the user demonstration, such as the order of moving objects or a preferred location of an object category, to a novel environment in a zero-shot manner, thereby eliminating the need to constantly provide task instructions. However, the above methods do not model object similarities, thereby limiting the scope of these approaches to template-like arrangement tasks (e.g., table setting, arranging an office desk). Additionally, these methods still require a demonstration for every new user or preference style.

Closest to our work is the collaborative filtering technique by Abdo et al. that learns user-specific preferences of grouping objects in containers as pairwise similarities between object categories [12]. This technique is also extended to both rearrange novel object categories and optimize probing for the preferences of a new user [12]. However, the agent

in Abdo et al. assumes that the type of organization being sought is known *a priori* (e.g., the robot knows to organize items by class, or by action affordances). Thus, for example, in putting away groceries or organizing a shelf, the user would always be required to specify the target organization type. Our work relaxes this assumption and does not assume the organization type is known *a priori*. Instead, our model infers the desired object similarities from contextual cues in the observed initial state. Additionally, the approach by Abdo et al. is limited to approaching rearrangement as modeling pairwise similarities between object instances while our proposed framework can model more general semantic similarities between objects. Finally, their method requires a mixture of experts in order to generalize to novel object categories while the proposed ConSOR framework only needs the ConceptNet labels of novel objects to perform zero-shot generalization.

Another approach to object rearrangement is to learn user-agnostic object placement preferences from crowdsourced object arrangements. Toris et al. present a multi-hypothesis model to learn to pick and place task templates representing the preferred placement locations of objects from human demonstrations [16]. Sarch et al. propose an embodied AI rearrangement framework that learns from commonsense object-location preferences in tidy households to identify misplaced objects in a novel home environment and move the object to the best matching receptacle [17]. In a similar work, Kant et al. contribute a benchmark and baseline for tidying household environments by identifying and rearranging misplaced objects without instruction using common-sense knowledge derived from a large language model [18]. Though the above rearrangement approaches successfully avoid task goal specifications, the methods only focus on modeling object-receptacle preferences and do not reason about pairwise object similarities when placing objects. In a different work, Wu et al. propose an imitation learning framework to learn the target distribution of desired object arrangements from expert examples as a gradient field [13].

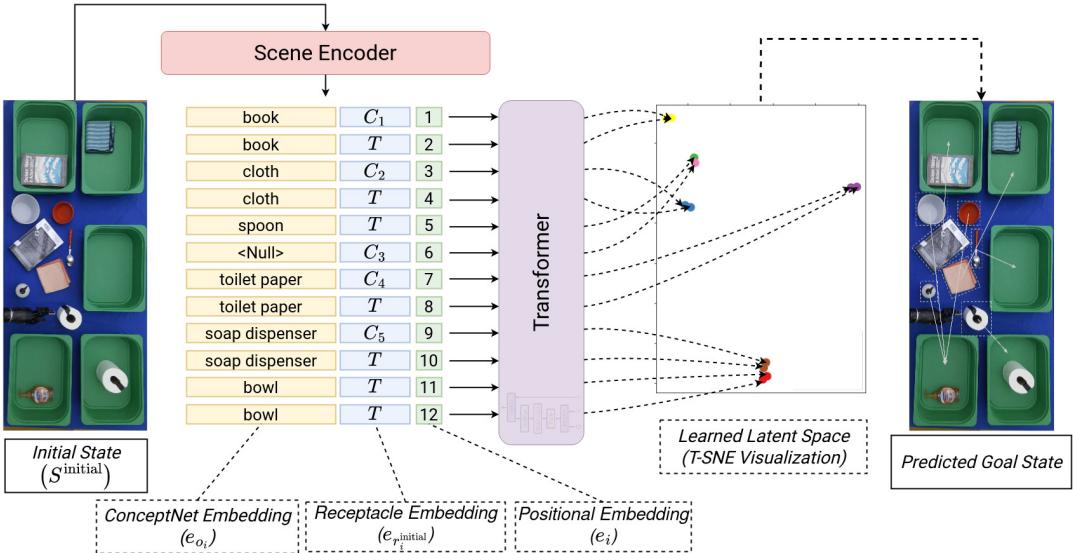


Fig. 2. Model architecture showing how ConSOR encodes a rearrangement scene and transforms it into a learned embedding space, which is then used to determine object placements in the predicted goal state. Each green bin represents a container receptacle  $c_i \in \mathcal{C}$ . In the initial state,  $\mathcal{X}_P$  is the set of objects already in a green bin, and  $\mathcal{X}_U$  is the set of objects on the blue table. (right) Predicted receptacle assignments for all objects in  $\mathcal{X}_U$  are shown using white arrows.

The learned target gradient field can then be used as a reward function to train a Reinforcement Learning (RL) agent to rearrange objects. However, the approach by Wu et al. requires separate models for different goal arrangement distributions and cannot identify the desired target distribution from the initial environment state, thereby restricting its usage to rearrangement tasks with only a single organizational style.

### III. OBJECT REARRANGEMENT OF PARTIALLY ARRANGED ENVIRONMENTS

We formalize object rearrangement in partially arranged environments as an instance of the general class of object rearrangement problems [1] in which the goal arrangement state is not explicitly stated to the robot, but instead must be inferred from the context provided by already arranged items in the scene. Specifically, the robot is presented with:

- a fixed set of receptacles  $\mathcal{R}$  in which objects can be placed (e.g., shelves, bins, containers),
- a set of prearranged objects,  $\mathcal{X}_P$ , arranged within the receptacles to match the user's intended organization schema (e.g., partially filled pantry where items are separated by meal type), and
- a set of unarranged objects,  $\mathcal{X}_U$ , for which the robot must find the correct receptacle in order to match the user's organizational schema.

Note that the user's schema is not explicitly specified and must be inferred from  $\mathcal{X}_P$  and  $\mathcal{R}$ . Multiple schemas can be applied to the same set of objects, and the robot's challenge is to infer the correct one and place  $\mathcal{X}_U$  accordingly.

At any given time, we model the state  $\mathcal{S}$  of the rearrangement environment as a set of tuples  $\{x_i, \dots, x_{N_x}\}$ , where  $N_x$  is the number of object instances in  $\mathcal{S}$ . Each object instance is represented as  $x_i = (o_i, r_i, i)$ , where  $o_i$  is the

object category/class,  $r_i \in \mathcal{R}$  is the receptacle in which  $o_i$  is placed, and  $i$  is an identifier index to distinguish object instances of the same category (e.g., multiple bowls in the same scene are assigned different values of  $i$  in the state representation  $\mathcal{S}$ ).  $\mathcal{R}$  contains a work surface  $T$  and the set of immovable containers  $\mathcal{C} = \{C_1, \dots, C_{N_C}\}$ , where  $N_C$  is the total number of containers in  $\mathcal{S}$ . Note that the containers in  $\mathcal{C}$  can generalize to different cabinets, shelves, or drawers in a real household. We represent a receptacle  $r_j$  that does not contain any object in  $\mathcal{S}$  by artificially placing a ‘null’ object in the receptacle and adding it to the state.

Given a partially arranged initial state  $\mathcal{S}^{initial}$ , our goal is to reach a desired goal state  $\mathcal{S}^{goal}$  by moving objects on  $T$  (i.e., objects in  $\mathcal{X}_U$ ) to containers  $\mathcal{C}$  such that the resulting arrangement matches the user's latent organizational schema. Note that our current problem does not consider visual semantics such as appearance similarities; however, this formulation can be extended to consider visual features by adding observation information to the state representation.

### IV. CONSOR TRANSFORMER MODEL ARCHITECTURE

To address the above problem, we introduce the Context-aware Semantic Object Rearrangement framework (ConSOR). ConSOR uses a learned Transformer encoder to generate an object-centric latent embedding space from the partially arranged initial state that mimics the object grouping in the desired goal state. The object-centric embeddings are then clustered to determine object placements in the predicted goal state. Prior work in robot rearrangement has combined object-centric state representations with the attention capabilities of Transformer encoders to enhance the generalization capabilities of these models to novel objects, scenes and tasks [10], [15], [19]. We adopt a similar approach in designing the encoder model of ConSOR and augment

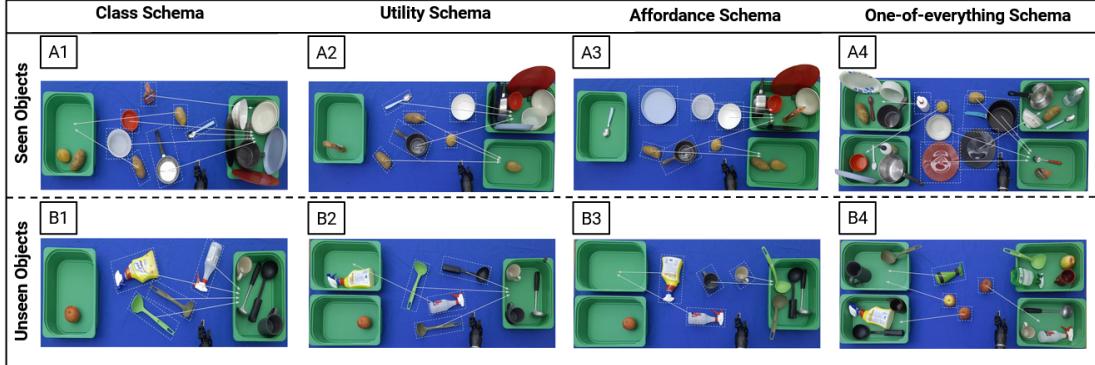


Fig. 3. Goal states for each schema from two different sets of objects. The top row of scenes are from the test dataset with seen object categories, while the bottom row of scenes are from the test dataset with unseen object categories.

object-centric representations with commonsense knowledge from ConceptNet to generalize to novel object categories.

Figure 2 presents a detailed structure of the ConSOR framework. Given the initial state  $\mathcal{S}^{initial}$ , we project each object instance  $x_i^{initial} = (o_i, r_i^{initial}, i)$  to a higher-dimensional space using a scene encoder  $h(x_i^{initial}) \rightarrow e_i$ , where  $e_i = [e_{o_i}, e_{r_i^{initial}}, e_i]$ . Specifically,  $e_{o_i}$  is the pre-trained ConceptNet Numberbatch vector [20] corresponding to the category  $o_i$ ,  $e_{r_i^{initial}}$  is a positional encoding to indicate which receptacle the object lies in, and  $e_i$  is a positional encoding of the indicator index  $i$ . The encoded scene  $\mathcal{V}^{initial} = \{h(x_1^{initial}), \dots, h(x_{N_x}^{initial})\}$  is passed through ConSOR’s Transformer encoder to output normalized latent embeddings  $\mathcal{L} = \{l_1, \dots, l_{N_x}\}$ . The encoder is trained with a triplet margin loss [21] to group embeddings of objects sharing the same receptacle in the goal state together and move embeddings of objects in different receptacles away from each other. In this manner, the encoder learns to generate a latent embedding space from the partially arranged initial state that mimics the object grouping in the desired goal state. During evaluation, ConSOR chooses the container to place each unarranged object instance  $x_i^U \in \mathcal{X}_U$  by calculating the centroid of each container in the latent space and choosing the container whose centroid has the highest cosine similarity with the corresponding latent vector  $l_i^U$ . Mathematically, the predicted placement  $\hat{r}_i^U$  of instance  $x_i^U$  is determined as

$$\hat{r}_i^U = \arg \max_{c \in \mathcal{C}} l_i^U \cdot l_c^{centroid} \quad (1)$$

where  $l_c^{centroid}$  is the latent centroid embedding of container  $c$  in  $\mathcal{S}^{initial}$ .

The encoder model of ConSOR consists of three stacked Transformer encoder layers, followed by an MLP layer to reduce the dimension of the generated embeddings and a  $L_2$  normalization layer. We train the model for 30 epochs (learning rate=1e-3, batch size=64, dropout=0.5) and perform early stopping based on the success rate obtained from evaluating on the validation dataset.

## V. DATASET OF ORGANIZATIONAL SCHEMAS FOR OBJECT REARRANGEMENT

To evaluate object rearrangement in partially arranged environments, we contribute a novel dataset of arranged scenes generated using household objects from the AI2Thor simulator [22]. To generate each scene, we defined four organizational schemas to determine how objects are grouped in the goal state:

- 1) Class schema ( $F_{class}$ ): grouping objects based on the affinity of their semantic concepts in WordNet [23],
- 2) Utility schema ( $F_{utility}$ ): grouping objects based on the affinity of their product categories mined from a popular retail store (Walmart [24]),
- 3) Affordance schema ( $F_{affordance}$ ): grouping objects with similar action affordance labels (6 affordance labels in total) gathered from the Moving Objects dataset [25], and
- 4) One-of-everything ( $FOOE$ ): distributing objects in containers such that each container holds exactly one of each object type (referred to as OOE for brevity).

We created a schema-balanced dataset by generating 1980 training, 110 validation, and 110 test goal scenes from each of the four schemas using a set of 28 object categories taken from the AI2Thor simulator [22] and grounded in WordNet. Example object categories include fruits, vegetables, office supplies, kitchen and dining accessories, cleaning supplies, bathroom accessories, and home decor. We also generated a secondary test dataset of 120 goal scenes using 10 novel object categories to test the generalization capability of ConSOR to object categories that were entirely unseen during training. Table I lists the objects present in our dataset. Partially arranged initial scenes are generated from goal scenes by sequentially removing randomly selected objects in  $\mathcal{X}^P$  from containers and placing them on work surface  $T$ . In this manner, we systematically vary the degree to which the presented organization is complete.

Figure 3 shows example arranged scenes from each schema. In the Class Schema (A1 and B1), objects are grouped by class similarity, such that vegetables should be placed in one bin, and kitchen items in the other. Note that

Seen Object Categories	Unseen Object Categories
Aluminum Foil, Basket Ball, Book, Bottle, Bowl, Bread, Candle, Cloth, Cup, Dish Sponge, Dumbbell, Egg, Hand Towel, Kettle, Laptop, Lettuce, Newspaper, Pen, Plate, Pot, Potato, Scrub Brush, Soap Dispenser, Spoon, Toilet Paper, Tomato, Towel, Wine Bottle	Apple, Box, Ladle, Mug, Pan, Paper Towel Roll, Pencil, Spray Bottle, Vase, Watering Can

TABLE I

TABLE OF 28 OBJECT CATEGORIES SEEN (LEFT) AND 10 UNSEEN (RIGHT) DURING TRAINING

in B1, the model is asked to generalize to cleaning supplies, with the goal of grouping them with kitchen items rather than vegetables due to more closely aligned similarity, as only two containers are provided in this example.

In the Utility Schema (A2, B2), the robot is provided with three containers. In A2, vegetables, a soap dispenser, and cooking supplies are organized into different bins. In B2, the model generalizes to previously unseen objects, placing cleaning supplies in their own bin.

In the Affordance Schema (A3, B3), objects are grouped by their afforded functionality. As this example highlights, detecting the desired organizational structure from a partial scene can be quite challenging. The key clue is given by the spoon, which is placed separately from other kitchen items. This is due to the spoon's shape (long handle with shallow convex hull) differing from that of the other objects (round with deep convex hull), thereby resulting in different affordances. Thus, the robot must learn to appropriately group the remaining items. Furthermore, note that scenes B2 and B3 have different initial states but end up in the same goal state; this type of aliasing makes the partial rearrangement problem complex, causes the differences between schemas to be less obvious, and requires that the learned model pay close attention to contextual cues in the initial state.

Finally, in the One-of-Everything Schema (A4, B4), the robot's objective is to place one of each object in the bins, akin to packing a lunch, or conference gift bags. We include this schema because, while appearing simple, it can pose quite a challenge to machine learning algorithms because similar objects must be separated rather than binned together. As we will show in the results, prior works struggle to find solutions to this schema.

Note that, although we define four types of schemas, ConSOR is trained only on the initial and goal states without any schema labels. Instead, ConSOR learns to distinguish between scenes of different schemas by learning the differences between them from the training data.

## VI. BASELINES AND METRICS

We compare ConSOR performance against two baselines:

**Abdo-CF:** *Abdo-CF* is a collaborative filtering technique proposed by Abdo et al. that learns user-specific pairwise object similarities from multiple preference ranking matrices of different users [12]. The learned pairwise preferences per user are then used to identify the placements of query objects via spectral clustering. Critically, this approach requires that

Prompt to GPT-3	Answers (Incorrect placements marked in red)
<p>There are 4 containers on the table. They contain the following objects:</p> <p>Container 1: Lettuce-0, Spoon-0, Wine Bottle-0  Container 2 is empty  Container 3: Lettuce-1, Spoon-1, Wine Bottle-1  Container 4: Lettuce-2, Plate-0, Pot-0, Spoon-2  The following objects are on the table: Lettuce-3, Plate-1, Plate-2, Plate-3, Pot-1, Pot-2, Pot-3, Spoon-3, Wine Bottle-2, Wine Bottle-3</p> <p>How would you place the objects on the table into the containers?</p>	<p>Container 1: Lettuce-0, Spoon-0, Wine Bottle-0, <u>Wine Bottle-2</u>  Container 2: Plate-1, <u>Plate-2</u>, <u>Plate-3</u>, Pot-1, <u>Pot-2</u>, <u>Pot-3</u>  Container 3: Lettuce-1, Spoon-1, Wine Bottle-1, <u>Wine Bottle-3</u>  Container 4: Lettuce-2, <u>Lettuce-3</u>, Plate-0, Pot-0, Spoon-2, <u>Spoon-3</u></p>

Fig. 4. Prompt given to GPT-3 and its response for a one-of-everything schema scene. The misplaced objects are marked in red underline.

the organization or schema type be known *a priori* and given as an input. By comparison, ConSOR implicitly infers the schema from contextual cues in the partially arranged initial scene. Additionally, *Abdo-CF* generalizes to unseen objects by relying on a mixture of experts providing object similarities of the unseen object, while ConSOR only requires the ConceptNet labels of unseen objects.

**GPT-3:** The Generative Pre-trained Transformer 3 (*GPT-3*) is an autoregressive language model that generates natural language in response to user input [14]. Prior works have demonstrated that large language models such as *GPT-3* are able to reason about sequential tasks and physical spaces [26], [27], [28]. To utilize *GPT-3* as a baseline, we prompt the model with a description of the partially arranged initial state along with one unlabeled demonstration from each schema to inform the language model about the desired output. Figure 4 shows a prompt taken from the test dataset and the corresponding response from GPT-3.

In our problem formulation, we seek to transform an initial object arrangement, represented by state  $\mathcal{S}^{initial}$ , to a goal object arrangement, represented by the goal state  $\mathcal{S}^{goal}$ , where the goal is not known to the robot *a priori* and must be inferred. We therefore evaluate object arrangement performance by measuring the similarity between the achieved object arrangement state and the goal state. To quantify this difference, we introduce a distance measure derived from *edit distance*, a widely used string similarity metric in computational linguistics [29]. Specifically, we define the Scene Edit Distance (*SED*) between states  $\mathcal{S}^A$  and  $\mathcal{S}^B$  as the minimum number of object displacements that must be made in  $\mathcal{S}^A$  to reach  $\mathcal{S}^B$ . In our problem formulation, this is equivalent to the number of misplaced objects in  $\mathcal{S}^A$  compared to  $\mathcal{S}^B$  and vice-versa.

Additionally, we derive two aggregate evaluation metrics from *SED* to measure rearrangement performance across an entire dataset. The first is the Success Rate,  $M^{SR}$ , which corresponds to the fraction of goal states predicted correctly:

$$M^{SR} = \frac{1}{D} \cdot \sum_{i=1}^D \mathbb{1}(SED(\hat{\mathcal{S}}_i, \mathcal{S}_i^{goal}) = 0) \quad (2)$$

where  $\hat{\mathcal{S}}_i$  is the predicted goal state,  $\mathcal{S}_i^{goal}$  the ground truth state for the initial state  $\mathcal{S}_i^{initial}$ , and  $D$  is the total number of examples in the test dataset.

The second metric is the Average Non-zero SED  $M^{NSED}$  or the average *SED* between incorrectly predicted goal

Method	Class Schema ( $F_{class}$ )		Utility Schema ( $F_{utility}$ )		One-of-everything Schema ( $F_{OOE}$ )		Affordance Schema ( $F_{affordance}$ )	
	$M^{SR}$	$M^{NSED}$	$M^{SR}$	$M^{NSED}$	$M^{SR}$	$M^{NSED}$	$M^{SR}$	$M^{NSED}$
ConSOR (Ours)	<b>99%</b>	<b>1.4 (SD=0.5)</b>	<b>99%</b>	<b>1.2 (SD=0.4)</b>	<b>100%</b>	-	<b>98%</b>	<b>1.0 (SD=0.0)</b>
Abdo-CF	89%	3.9 (SD=1.3)	93%	3.6 (SD=0.8)	0%	15.4 (SD=3.1)	90%	3.4 (SD=1.1)
GPT-3	36%	3.1 (SD=2.1)	41%	3.2 (SD=2.2)	4%	9.8 (SD=5.6)	40%	3.3 (SD=2.2)

TABLE II

EVALUATION RESULTS FOR EACH SCHEMA CALCULATED ON TEST DATA OF REARRANGEMENT SCENES WITH UNSEEN OBJECT ARRANGEMENTS.

Method	Class Schema ( $F_{class}$ )		Utility Schema ( $F_{utility}$ )		One-of-everything Schema ( $F_{OOE}$ )		Affordance Schema ( $F_{affordance}$ )	
	$M^{SR}$	$M^{NSED}$	$M^{SR}$	$M^{NSED}$	$M^{SR}$	$M^{NSED}$	$M^{SR}$	$M^{NSED}$
ConSOR (Ours)	<b>99%</b>	<b>1.0 (SD=0.0)</b>	<b>91%</b>	<b>1.3 (SD=0.5)</b>	<b>100%</b>	-	<b>97%</b>	<b>1.0 (SD=0.0)</b>
GPT-3	40%	3.3 (SD=2.2)	61%	2.9 (SD=2.2)	3%	9.2 (SD=5.4)	44%	3.2 (SD=2.1)

TABLE III

EVALUATION RESULTS FOR EACH SCHEMA CALCULATED ON TEST DATA OF REARRANGEMENT SCENES WITH NOVEL OBJECT CATEGORIES.

states ( $SED > 0$ ) and their ground truth states. This is defined as:

$$M^{NSED} = \frac{\sum_{i=1}^D SED(\hat{S}_i, \mathcal{S}_i^{goal}) \cdot \mathbb{1}(SED(\hat{S}_i, \mathcal{S}_i^{goal}) > 0)}{\sum_{i=1}^D \mathbb{1}(SED(\hat{S}_i, \mathcal{S}_i^{goal}) > 0)} \quad (3)$$

Together, the above two metrics capture a model’s performance, such that  $M^{SR}$  reports the percentage of arrangements that an algorithm gets completely right, and  $M^{NSED}$  reports the degree of dissimilarity for scenes that were not correct (non-zero SED).

## VII. EVALUATION RESULTS

In this section, we present results of two generalization experiments, first evaluating generalization to previously unseen arrangements with known objects, and second evaluating zero-shot generalization to novel object categories. Additionally, we present insights characterizing the differences in learned embedding spaces across schemas, and evaluate the effect of training data size on performance.

### A. Generalizing to Unseen Object Arrangements

Table II presents a summary of evaluation results from testing on scenes with unseen object arrangements. ConSOR yields a higher  $M^{SR}$  than both Abdo-CF and GPT-3 across all four schemas. Notably, our framework is also the only method in our evaluation to perfectly rearrange  $FOOE$  scenes. Additionally, ConSOR has the least  $M^{NSED}$  score across all four schemas, indicating that, in the rare cases that errors occur, ConSOR generates state predictions that are closer to the true goal state than the baseline approaches. Abdo-CF has the second-best performance in three out of four of the schemas while failing to successfully rearrange a single  $FOOE$  scene. We attribute the failure in  $FOOE$  by Abdo-CF to the inductive bias of collaborative filtering, as it is difficult to mimic the  $FOOE$  schema using pairwise object similarities. GPT-3 performs the worst on three out of four schemas with a slightly higher  $M^{SR}$  in  $FOOE$  than Abdo-CF. This shows that the general-purpose commonsense knowledge learned by GPT-3 is insufficient to model an organizational schema with a specific set of

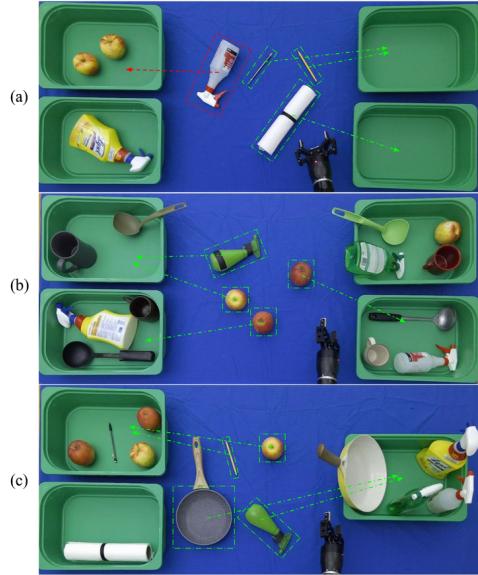


Fig. 5. Predicted goal states generated by ConSOR for scenes with novel object categories. Correct object placements are shown using a dash-dotted green bounding box and arrow, while incorrect object placements are shown using a dashed red bounding box and arrow.

semantic constraints, thus necessitating the need for our proposed framework.

### B. Zero-Shot Generalization to Novel Object Categories

Table III shows our evaluation results from testing on scenes with novel object categories. We do not evaluate the baseline Abdo-CF on novel object categories as this method requires external semantic knowledge to rearrange objects unseen during training, and the lack of this knowledge with other methods leads to an unfair comparison. ConSOR is able to successfully leverage the commonsense knowledge embedded in ConceptNet to perform zero-shot generalization to completely novel scenes and outperform GPT-3. Also, in comparison to our model’s performance on unseen object arrangements, ConSOR retains performance on three out of four schemas, with  $F_{utility}$  showing the largest drop in performance. We believe this drop in performance is due to the Utility schema deviating the most from the commonsense knowledge embedded in ConceptNet.

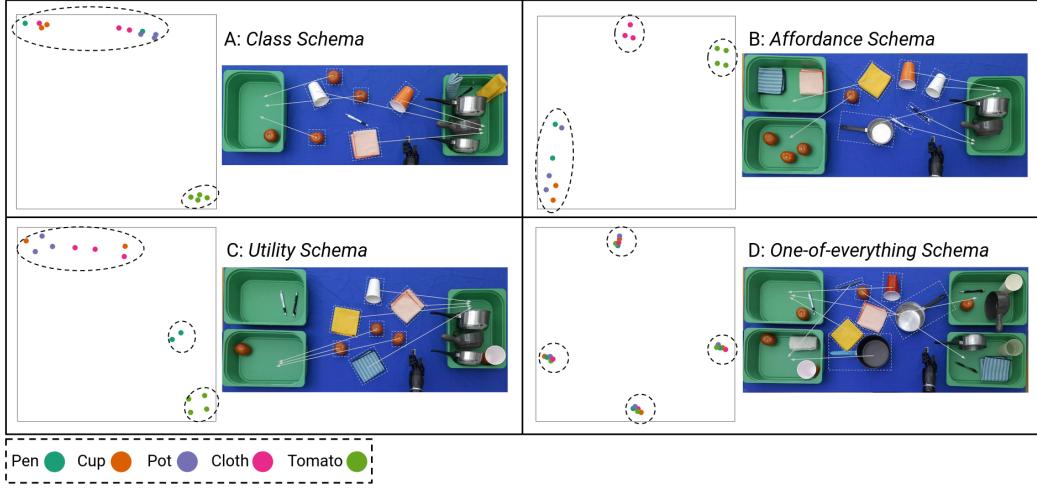


Fig. 6. Visualizing the learned embedding space of ConSOR for scenes of different schemas. The right side of each cell shows the initial scene and ground truth object placements, and the left side is a T-SNE projection of the generated object-centric embeddings in two dimensions.

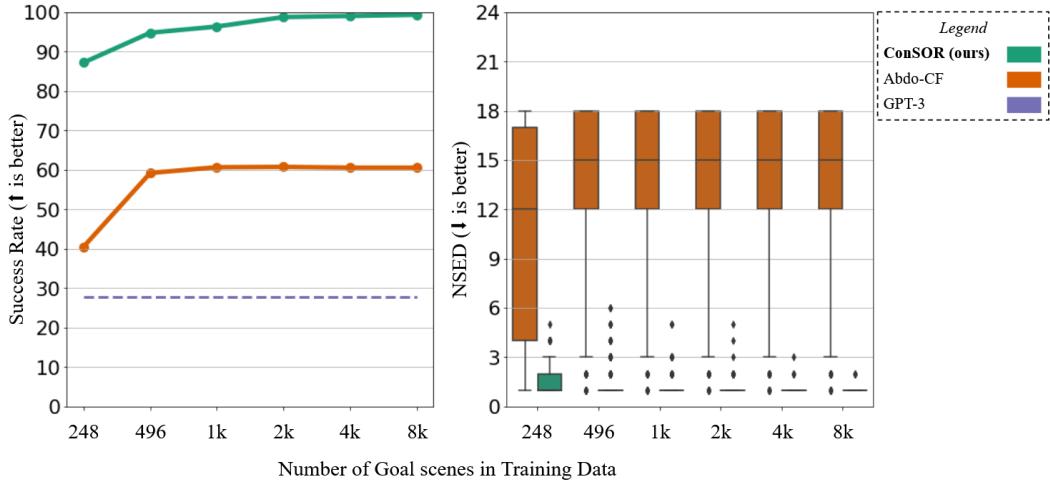


Fig. 7. Measuring the change in performance of each rearrangement method based on the number of goal scenes in the training data. The plot on the left is a line diagram of success rate averaged across all schemas versus the size of training data, with the success rate of *GPT-3* shown as a dotted line. The plot on the right is a series of box plots showing the varying range of non-zero *SED* scores for ConSOR and *Abdo-CF* with an increase in training data.

Figure 5 presents some of the correct and incorrect goal predictions made by ConSOR for scenes with novel object categories. We observe in Figure 5(a) that ConSOR occasionally places objects of the same category in separate containers even when the desired goal schema is not *FOOE*. We hypothesize that this may be attributed to the model lacking confidence about the desired schema, resulting in a goal state with a ‘hybrid’ schema. Figures 5(b) and 5(c) show two accurate goal predictions, belonging to *FOOE* and *Futility* respectively. In both, ConSOR leverages contextual cues about the desired schema from the initial state, such as the number of containers in the scene and the current object arrangement, to perfectly generate the goal state.

#### C. Visualizing the difference in scene organization across schemas in learned embedding space

Evaluation results in previous sections show that ConSOR successfully learns a mapping from the initial partially ar-

ranged rearrangement state to a latent space of embeddings mimicking the desired object grouping. We visualize the embeddings generated by our method for four different initial state configurations from the test dataset in Figure 6 using a T-SNE projection [30]. We observe that, for the same set of object categories, the learned embedding space of ConSOR adapts itself to the structure of the scene (number of containers) as well as the organization of objects in the initial state. For example, the latent spaces of scenes A (Class schema) and D (One-of-everything Schema) highlight the differences in the two scenes, namely the different numbers of containers in the scene and whether the same object categories are grouped together or separately. On the other hand, scenes B (Affordance schema) and C (Utility schema) can only be differentiated by observing whether ‘Pen’ or ‘Cloth’ is grouped with ‘Pot’, and this is seen in the latent space as well. We also note that ConSOR is able to identify the differences between scenes of different

schemas without being explicitly trained with the actual schema labels, and instead learns to differentiate schemas from unlabelled training data.

#### D. Effect of Size of Training Data on Performance

Finally, we evaluate the effect of training data size on the performance of ConSOR and *Abdo-CF*. Figure 7 shows the change in average  $M^{SR}$  and  $M^{NSED}$  values for different numbers of goal scenes in the training data. All metrics were calculated using the same test dataset as the previous sections. We observe the average  $M^{SR}$  of ConSOR steadily rising and the variance of  $M^{NSED}$  scores decreasing with more training data, while *Abdo-CF* performance saturates after 496 training goal scenes. Across all training data sizes, we find that ConSOR has a higher average  $M^{SR}$  and lower mean  $M^{NSED}$  score than *Abdo-CF*.

### VIII. CONCLUSION AND DISCUSSION

This work introduces ConSOR, a semantic reasoning framework for object rearrangement. ConSOR relies on contextual cues from a partially arranged environment to infer the desired goal state by generating a learned object-centric latent space that mimics the arrangement in the desired goal state. Additionally, ConSOR leverages external commonsense knowledge from ConceptNet to perform zero-shot generalization to rearrange scenes with novel object categories. We evaluated our proposed framework on a dataset of 8k arranged scenes, each belonging to one of four ‘organizational schemas’, and found that our approach strongly outperforms both the *Abdo-CF* and *GPT-3* baseline across all tested conditions.

Note that ConSOR outperforms the next leading baseline, *Abdo-CF*, even though the baseline is explicitly given the target schema type as input (e.g., *class schema*) while ConSOR is required to infer this information automatically from context. In a real-world setting, ConSOR would therefore require significantly less effort from the user.

One assumption of our approach is that the robot knows which items have been pre-arranged, and thus serve as useful context, and which still need to be put away. Depending on the real-world application, making this distinction may be simple or quite difficult (e.g., bags of new groceries are easy to detect, out-of-place living room objects pose a greater challenge). Recent work on visuo-semantic commonsense priors may help inform this decision in the future [17], [18].

### REFERENCES

- [1] D. Batra, A. X. Chang, S. Chernova, A. J. Davison, J. Deng, V. Koltun, S. Levine, J. Malik, I. Mordatch, R. Mottaghi et al., “Rearrangement: A challenge for embodied AI,” *arXiv preprint arXiv:2011.01975*, 2020.
- [2] C. Paxton, C. Xie, T. Hermans, and D. Fox, “Predicting stable configurations for semantic placement of novel objects,” in *CoRL*. PMLR, 2022, pp. 806–815.
- [3] C. Wang, D. Xu, and L. Fei-Fei, “Generalizable task planning through representation pretraining,” *IEEE RAL*, vol. 7, no. 3, 2022.
- [4] A. Goyal, A. Mousavian, C. Paxton, Y.-W. Chao, B. Okorn, J. Deng, and D. Fox, “IFOR: Iterative flow minimization for robotic object rearrangement,” in *CVPR*, 2022, pp. 14 787–14 797.
- [5] H. Wu, J. Ye, X. Meng, C. Paxton, and G. S. Chirikjian, “Transporters with visual foresight for solving unseen rearrangement tasks,” in *IROS*. IEEE, 2022, pp. 10 756–10 763.
- [6] W. Goodwin, S. Vaze, I. Havoutis, and I. Posner, “Semantically grounded object matching for robust robotic scene rearrangement,” in *ICRA*. IEEE, 2022, pp. 11 138–11 144.
- [7] W. Liu, Y. Du, T. Hermans, S. Chernova, and C. Paxton, “StructDiffusion: Language-guided creation of physically-valid structures using unseen objects,” in *RSS*, 2023.
- [8] M. Shridhar, L. Manuelli, and D. Fox, “Perceiver-Actor: A multi-task transformer for robotic manipulation,” in *CoRL*, 2022.
- [9] —, “CLIPort: What and where pathways for robotic manipulation,” in *CoRL*, 2021.
- [10] V. Jain, Y. Lin, E. Undersander, Y. Bisk, and A. Rai, “Transformers are adaptable task planners,” in *CoRL*, 2023.
- [11] I. Kapelyukh and E. Johns, “My House, My Rules: Learning tidying preferences with graph neural networks,” in *CoRL*. PMLR, 2022, pp. 740–749.
- [12] N. Abdo, C. Stachniss, L. Spinello, and W. Burgard, “Organizing objects by predicting user preferences through collaborative filtering,” *IJRR*, vol. 35, no. 13, pp. 1587–1608, Nov. 2016.
- [13] M. Wu, F. Zhong, Y. Xia, and H. Dong, “TarGF: Learning target gradient field to rearrange objects without explicit goal specification,” in *NeurIPS*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 31 986–31 999.
- [14] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell et al., “Language models are few-shot learners,” *NeurIPS*, vol. 33, pp. 1877–1901, 2020.
- [15] Y. Jiang, A. Gupta, Z. Zhang, G. Wang, Y. Dou, Y. Chen, L. Fei-Fei, A. Anandkumar, Y. Zhu, and L. Fan, “VIMA: General robot manipulation with multimodal prompts,” in *ICML*, 2023.
- [16] R. Toris, D. Kent, and S. Chernova, “Unsupervised learning of multi-hypothesized pick-and-place task templates via crowdsourcing,” in *ICRA*. IEEE, 2015, pp. 4504–4510.
- [17] G. Sarch, Z. Fang, A. W. Harley, P. Schydlo, M. J. Tarr, S. Gupta, and K. Fragkiadaki, “TIDEE: Tidying up novel rooms using visuo-semantic commonsense priors,” in *ECCV*, 2022.
- [18] Y. Kant, A. Ramachandran, S. Yenamandra, I. Gilitschenski, D. Batra, A. Szot, and H. Agrawal, “Housekeep: Tidying virtual households using commonsense reasoning,” in *ECCV*. Springer, 2022, pp. 355–373.
- [19] W. Liu, C. Paxton, T. Hermans, and D. Fox, “StructFormer: Learning spatial structure for language-guided semantic rearrangement of novel objects,” in *ICRA*. IEEE, 2022, pp. 6322–6329.
- [20] R. Speer, J. Chin, and C. Havasi, “ConceptNet 5.5: An open multilingual graph of general knowledge,” in *AAAI*, 2017, p. 4444–4451.
- [21] X. Dong and J. Shen, “Triplet loss in siamese network for object tracking,” in *ECCV*, 2018, pp. 459–474.
- [22] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, M. Deitke, K. Ehsani, D. Gordon, Y. Zhu et al., “AI2-THOR: An interactive 3d environment for visual AI,” *arXiv preprint arXiv:1712.05474*, 2017.
- [23] G. A. Miller, “WordNet: A lexical database for english,” *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.
- [24] “Walmart.” [Online]. Available: <http://www.walmart.com/>
- [25] R. Goyal, S. Ebrahimi Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag et al., “The ‘Something Something’ video database for learning and evaluating visual common sense,” in *ICCV*, 2017, pp. 5842–5850.
- [26] L. Logeswaran, Y. Fu, M. Lee, and H. Lee, “Few-shot subgoal planning with language models,” in *NAAACL*. ACL, 2022, pp. 5493–5506.
- [27] J. R. Kirk, R. E. Wray, P. Lindes, and J. E. Laird, “Integrating diverse knowledge sources for online one-shot learning of novel tasks,” *arXiv preprint arXiv:2208.09554*, 2022.
- [28] A. Brohan, Y. Chebotar, C. Finn, K. Hausman, A. Herzog, D. Ho, J. Ibarz, A. Irpan, E. Jang, R. Julian et al., “Do As I Can, Not As I Say: Grounding language in robotic affordances,” in *CoRL*, 2022.
- [29] V. I. Levenshtein et al., “Binary codes capable of correcting deletions, insertions, and reversals,” in *Soviet physics doklady*, vol. 10, no. 8. Soviet Union, 1966, pp. 707–710.
- [30] L. Van der Maaten and G. Hinton, “Visualizing data using T-SNE,” *Journal of Machine Learning Research*, vol. 9, no. 11, 2008.