PARSEC: Preference Adaptation for Robotic Object Rearrangement from Scene Context

Kartik Ramachandruni and Sonia Chernova

***Abstract*— Object rearrangement is a key task for household robots requiring personalization without explicit instructions, meaningful object placement in environments occupied with objects, and generalization to unseen objects and new environments. To facilitate research addressing these challenges, we introduce PARSEC, an object rearrangement benchmark for learning user organizational preferences from observed scene context to place objects in a partially arranged environment. PARSEC is built upon a novel dataset of 110K rearrangement examples crowdsourced from 72 users, featuring 93 object categories and 15 environments. We also propose ContextSortLM, an LLM-based rearrangement model that places objects in partially arranged environments by adapting to user preferences from prior and current scene context while accounting for multiple valid placements. We evaluate ContextSortLM and existing personalized rearrangement approaches on the PARSEC benchmark and complement these findings with a crowdsourced evaluation of 108 online raters ranking model predictions based on alignment with user preferences. Our results indicate that personalized rearrangement models leveraging multiple scene context sources perform better than models relying on a single context source. Moreover, ContextSortLM outperforms other models in placing objects to replicate the target user's arrangement and ranks among the top two in all three environment categories, as rated by online evaluators. Importantly, our evaluation highlights challenges associated with modeling environment semantics across different environment categories and provides recommendations for future work.**

I. Introduction

Consider a robot that assists users with tidying the home by putting away objects, or what is known as the rearrangement problem [1]. How should this robot determine the appropriate location to put each object? The robot must develop an object placement strategy based on a user's organizational preferences without detailed instructions or demonstrations, as doing so would unnecessarily burden that user and would need to be repeated for new objects or environments. Furthermore, the robot should ensure that its actions align with the current arrangement of objects in the home (e.g., arrange new pantry items in accordance with existing ones.) Critically, these capabilities must apply to previously unseen objects and new homes. Thus, personalized object rearrangement presents three practical challenges: inferring user preferences without explicit instruction to determine the desired goal, meaningfully placing objects in pre-occupied environments, and adapting to unseen objects and new environments.

Prior work has proposed various approaches to infer rearrangement preferences without explicit instruction [2], [3], [4], [5], [6], [7], [8]. Most methods rely on either *prior* scene context [2], [3], [4] derived from prior observations of the user arranging objects, and *within-scene* context [5], [6] from the positions of objects already placed in partially arranged environments, to learn user preferences. However, *observation*-based rearrangement models assume that the current environment is empty, and *within-scene* based models do not perform well in environments sparsely occupied with objects. Few methods combined both context sources [7], [8], but either require additional user interaction to infer preferences [8] or fail to handle unseen objects and new environments [7]. Overall, approaches in prior work do not address all the aforementioned challenges.

To facilitate more research in personalized rearrangement, we present PARSEC, an object tidying benchmark and dataset addressing user personalization, object placement in partially arranged environments, and generalization to unseen objects and new environments. In this benchmark, a robot learns user preferences by leveraging *prior* and *within-scene* semantic context to determine object placements in environments occupied with objects. Existing datasets fall short of our objective, either due to lack of real-user data [9], [6], omission of user preferences [10], or limited objects and environments [11], [12]. To address this, we collect a novel crowdsourced dataset of real users arranging various household objects to complete organizational tasks, such as stocking the kitchen pantry and organizing the fridge. We evaluate existing rearrangement approaches across different environment types and initial conditions in PARSEC and complement these findings with a crowdsourced user evaluation, where online raters rank model predictions based on alignment with the target user's preference. Addressing the gap in prior work, we also propose a Large Language Model (LLM)-based approach that integrates scene context from both *prior* and *within-scene* context to place objects in partially arranged environments meaningfully.

Our work makes the following contributions. First, we formalize the problem of personalized rearrangement in partially arranged environments. Second, we introduce PARSEC as an evaluation framework for the above problem. The benchmark features a novel dataset of 110K rearrangement examples collected from 72 real users, covering 93 household objects across 15 environment instances [1]. Our dataset captures more diverse and flexible preferences than prior rule-based personas, as discussed in Section IV. Third, we propose ContextSortLM, an LLM-based personalized rear-

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

[1]The code and data will be made available at https://github.com/kartikvrama/parsec.

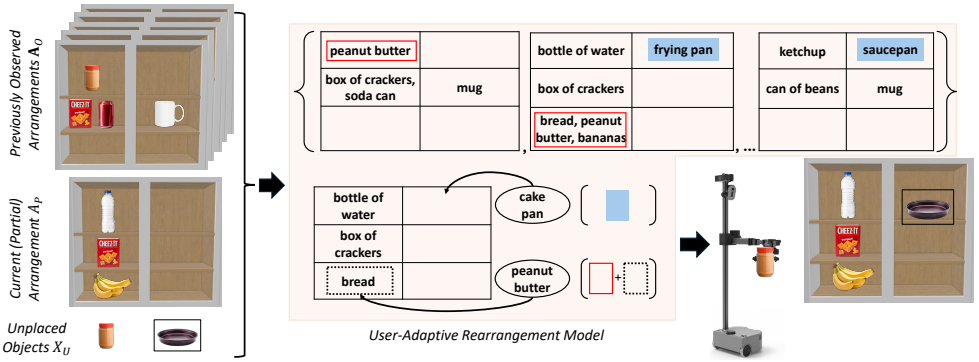


Fig. 1: In the PARSEC benchmark, the robot adapts to a user's organizational preferences when placing new objects $\mathcal{X}_U$ in an partially arranged, or pre-occupied, environment. The robot must jointly reason across prior observations of the user arranging objects, $\mathbf{A}_O$, and the environment's current object arrangement, $\mathcal{A}_P$, to meaningfully place objects.

rangement model that encodes *prior* scene context from multiple observations into a structured preference representation, explicitly accounting for multiple valid placements, to then place objects in a partially arranged environment without disrupting the environment's current arrangement. Lastly, we evaluate ContextSortLM and existing personalized rearrangement approaches on the PARSEC benchmark, complemented by a crowdsourced user evaluation with 108 online raters assessing the alignment of different model predictions to user preferences. Our combined results demonstrate the superior performance of models integrating multiple scene context sources for personalized object placement while revealing challenges in modeling environment semantics, which can lead to discrepancies in inferred user preferences. Moreover, ContextSortLM outperforms other models in computational evaluations in all three environment categories and ranks among the top two in all three environment categories, as rated by online evaluators, emphasizing the advantage of its structured preference representation. We summarize the key takeaways from our evaluation experiments in Section VIII to guide future work.

II. Problem Formulation

We formalize personalized object rearrangement in partially arranged environments as an extension of the rearrangement problem formulation in previous work [6]. Given an environment with placeable surfaces denoted by the set $\mathcal{S}$, we define an *arrangement* as a set of object-surface pairs representing the contents of each surface: $\mathcal{A} = \{(x, s)\}$, where x denotes an object placed on surface s. Object x is not confined to a predetermined closed set and is represented using natural language labels. The set of surfaces $\mathcal{S}$ is fixed for a single environment but may change across environments.

In this task, a robot must adapt to a user's object rearrangement preferences by relying on observation instead of explicit user input to then place objects in an environment occupied with objects. The robot's goal is to place a new set of objects $\mathcal{X}_U$ on surfaces S to transform the environment from its current arrangement $\mathcal{A}_P$ into $\mathcal{A}_G$. The environment may be empty ($\mathcal{A}_P = \Phi$) or partially arranged, meaning that it already contains a few objects. Optionally, the robot has access to prior observations of the environment, denoted as $\mathbf{A}_O = \{\mathcal{A}_O^1, \mathcal{A}_O^2, \ldots, \mathcal{A}_O^N\}$, which may include various object configurations. The length of observation history N depends on the rearrangement technique; many rearrangement models [4], [7], [12], [13] are not limited by the number of observations, but some approaches only consider a single observation ($N = 1$) [9], [3] or do not use any prior observations ($N = 0$) [6], [5]. Similarly, some rearrangement techniques [4], [12], [9] do not model the environment's current arrangement and assume the environment is empty, or $\mathcal{A}_P = \Phi$.

The robot has access to two sources of semantic scene context. *prior* scene context is derived from passive observations of the user arranging objects, or $\mathbf{A}_O$, and *within-scene* context is obtained from the placement of objects in the initial state of the environment, or $\mathcal{A}_P$. Figure 1 illustrates how a robotic agent would combine both context sources to place new objects. The core challenge in this problem is attending to specific contextual cues from *prior* and *within-scene* context to reason about the placement of each unplaced object. For example, in Figure 1, the peanut butter's placement is ambiguous because it was observed on multiple surfaces (marked with a red box). The peanut butter is eventually placed on the bottom-left shelf, next to the bread (black dotted box), as prior observations show the user groups sandwich ingredients and snacks together. Rearrangement models must jointly reason across both context sources to meaningfully place objects in partially arranged environments.

III. Prior Work in Personalized Rearrangement

In this section, we summarize prior work in personalized rearrangement approaches, categorizing models based on their input modality and reliance on *prior* or *within-scene* context. We also assess their generalization capabilities to new objects and environments and discuss relevant datasets for our benchmark.

A. Vision and Graph Representations for Personalized Rearrangement

Prior work on inferring user preferences from observed object arrangements falls under one of two categories, based on

the modality used to represent object arrangements: *vision-based rearrangement models* that infer object placement directly from visual observations [5], [3], [2], and *graph-based rearrangement models* that determine object placement from abstracted scene graphs of object arrangements [6], [12], [9], [4], [7], [14]. The choice of modality influences the precision of object placement and generalization capabilities to new objects and unseen environments.

Vision-based rearrangement models infer object placements directly from RGB or RGB-D observations. These techniques achieve precise object placement by predicting geometrical coordinates, either through a neural vision-to-placement policy [5], a vision-language model [2] or a search algorithm utilizing semantic-geometric maps [3]. Vision-based approaches also adapt to new user preferences, either through *prior* [2], [3] or *within-scene* [5] semantic context. However, these methods have some trade-offs. Trabucco et al. [3] cannot rearrange objects unobserved in prior observations, and Ramrakhya et al. [5] cannot rearrange new objects. Newman et al. [2] generalizes to new objects through vision-language models, but is restricted to performing table-top rearrangement and does not generalize to more complex environments.

Graph-based rearrangement approaches represent object arrangements as scene graphs, where nodes correspond to objects and placeable surfaces, and edges represent object-surface placements. These methods infer object placement from scene graph through neural network rearrangement policies [12], [6], [14] or by transforming into alternate representations, such as text [9], [8] or pairwise ranking matrices [4], [7]. Graph-based rearrangement models generalize to unseen objects by leveraging semantic similarities among objects, either through explicit knowledge graphs [4], [7] or by using pre-trained semantic embeddings [12], [6]. However, most graph-based approaches trade off precise placement, typically assigning objects to predefined surfaces rather than outputting exact coordinates. Moreover, some methods either disregard semantic information from the environment [6], [4], or overfit to a single environment [12], [7]. Wu et al. [9] and Wang et al. [8] provide notable exceptions by incorporating LLMs, enabling generalization to new environments.

While vision and graph-based rearrangement models have complementary strengths in adapting to new users, graph-based models excel at generalizing to unseen objects – an essential capability for robots working in human environments. Since directly comparing these model types is challenging, we will focus exclusively on graph-based approaches for the remainder of the paper.

Table I compares existing graph-based rearrangement methods based on their use of semantic context for preference adaptation and the semantic information encoded by each model. Few rearrangement models use both *prior* and *within-scene* context for preference adaptation [7], [8], [14]. However, Brawner and Littman [7] fail to handle unseen objects and new environments, and Wang et al. [8] require user interaction during each rollout to disambiguate *prior* scene context. Sarch et al. [14] also leverages *prior* and *within-scene* context to determine plausible object-receptacle pairs, but memorize a single user's preferences during training and cannot adapt to new user preferences without retraining. Our proposed approach, ContextSortLM, adapts to new user preferences by integrating *prior* and *within-scene* context to place objects in partially arranged environments without user supervision. ContextSortLM also leverages LLM reasoning to generalize to unseen objects and new environments. We present the details of this model in Section V.

	Scene Context		Models Semantics about X	
Rearrangement Model	**Observed Arrangements**	**Current Environment**	**Environment**	**Objects**
CF [4]	✓			✓
NeatNet [12]	✓			✓
TidyBot [9]	✓		✓	✓
ConSOR [6]		✓		✓
CF+ [7]	✓	✓		
APRICOT [8]	✓	✓	✓	✓
ContextSortLM (Ours)	✓	✓	✓	✓

TABLE I: Comparison of existing rearrangement approaches that adapt to new user preferences, categorized by their use of semantic context for preference adaptation and the semantic information encoded by the rearrangement model. Our proposed model, ContextSortLM, integrate scene context from prior and current observations via a unique preference representation to ensure user-aligned object placements.

B. Personalized Rearrangement Datasets

Existing datasets for personalized object rearrangement include rule-based [6], [9] and user-generated [12], [2] datasets of object arrangements. Rule-based datasets [6], [9] are generated from pre-defined organizational rules that dictate object placement and grouping, such as 'organize objects by their affordance' [6] or 'put shirts on the sofa and other clothes in the closet' [9]. These rule-based datasets scale well, but lack the nuances and diversity of real user preferences, as we demonstrate in Section IV. Alternately, user-generated datasets [12], [2] collected by online workers performing grounded organizational tasks [13], [11], [12], capture diverse placement preferences and record fine-grained placements. However, the data is collected for very specific organizational tasks in 1-2 fixed environments and contains a limited number of object categories and environments. Addressing the limitations of prior datasets, we collect a crowdsourced dataset of 432 object arrangements from real users that spans 5 different organizational tasks and involves various object categories and environments.

IV. PARSEC Benchmark and Dataset

For the PARSEC benchmark, we seek a dataset that a) incorporates data from real people instead of rules, b) includes multiple examples for each user, and c) spans a diverse set of object categories and environments. Since existing datasets do not meet our criteria, we collected our own data by hiring online workers to arrange household objects in environments resembling real homes [2]. Each online worker separately arranged six sets of objects in a single environment accompanied with short descriptions. The object sets were sampled to include objects relevant to the environment type and random household objects, and workers were instructed to only arrange the objects they considered relevant.

[2]Data collection was IRB exempt. Workers were hired from Prolific.

Fig. 2: The environments in the PARSEC benchmark can be categorized by the number of surface types and their position. A and B illustrate some examples of real user arrangements from the dataset.

In total, we collected 432 object arrangements, involving 93 household objects and spanning 72 users [3] and 15 environment instances. We uniformly sampled the environments from five household organizational tasks: stocking a kitchen pantry, arranging a bathroom cabinet, rearranging a bedroom dresser, stocking a fridge, and decorating a display shelf. These environments fall under three semantic categories, as shown in Figure 2: Environments with identical surface types and surfaces positioned vertically, or `Similar-1D`; environments with identical surface types, arranged in a 2D configuration, or `Similar-2D`; and environments with more than one surface type, or `Dissimilar`.

Each category presents unique challenges for adapting to user preferences. In environments with multiple identical surfaces (`Similar-1D` and `Similar-2D`), users pay more attention to which objects to group than the exact surface to place them. For instance, in example A of Figure 2, a user may prioritize grouping all types of mugs but place them in the left or right shelves interchangeably. In contrast, environments with multiple surface types (`Dissimilar`) encourage users to assign objects to specific surfaces. However, these preferences become more nuanced when multiple instances of each surface type exist. For example, in example B, users assigns objects to separate surfaces (e.g., self-care items on the table) while maintaining specific organizational patterns among identical surfaces (e.g., mementos on the left shelf and baskets on the right). Modeling the relative positions of surfaces in two dimensions (`Similar-2D` and `Dissimilar`) adds further complexity when generalizing to unseen environments.

Dataset Generation: From the crowdsourced object arrangements, we created a dataset of 110K examples. Each user annotator m provided six object arrangements within the same environment instance. Each object arrangement $\mathcal{A}_i = \{(x, s)\}$ comprises objects x, represented by text labels of its semantic category, and surfaces s, described as a tuple of surface type and relative 2D position. Optionally, s can be expressed as a templated language description, such as 'top-right shelf'. To create rearrangement examples, we iteratively selected one arrangement as the target arrangement $\mathcal{A}^*_G = \mathcal{A}_i$ and designated the other five as observed user arrangements

[3]Out of 75 users, three were removed due to failed attention checks.

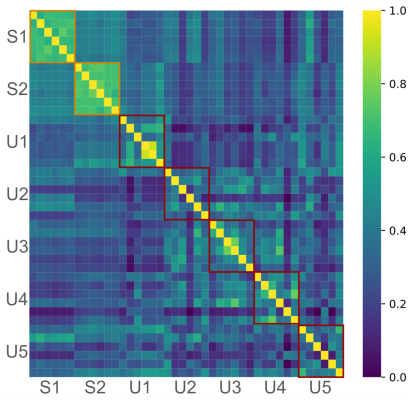


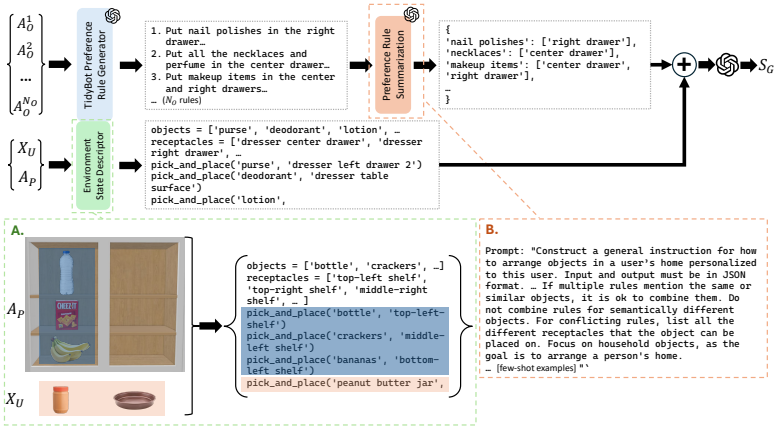
Fig. 3: To demonstrate the within-user variability and across-user diversity of crowdsourced rearrangement data, we plot the average WordNet similarity among both rule-based (S1, S2) and real user object arrangements (U1-U5) of a fridge environment.

$\mathbf{A}_O = \{\mathcal{A}_j | j \neq i\}$, generating $\binom{5}{2}$ pairs of $(\mathbf{A}_O, \mathcal{A}^*_G)$. For each target arrangement $\mathcal{A}^*_G$, we created several pairs of partially arranged states $\mathcal{A}_P$ and unplaced object sets $\mathcal{X}_U$ by randomly omitting objects from $\mathcal{A}^*_G$, labeling the omitted objects as the unplaced set. Empty environment states were also constructed by removing all objects from $\mathcal{A}_P$. The resulting dataset is represented as a set of tuples: $\mathcal{D} = \{(m, \mathbf{A}_O, \mathcal{A}_P, \mathcal{X}_U)\}$. Given $\mathbf{A}_O$ and $\mathcal{A}_P$, the rearrangement model must place $\mathcal{X}_U$ alongside $\mathcal{A}_P$ while adhering to m's preferences.

Comparing Manually Defined and Real User Preferences: We compared our crowdsourced data from a fridge in the `Dissimilar` category with two rule-based user personas resembling the sorting criteria from [9], with preference rules such as 'Put soda cans and eggs on the top door shelf' and 'Put produce on the middle shelf'. Figure 3 visualizes the average WordNet similarity between rule-based arrangements (S1, S2) and real user arrangements (U1–U5). We find that real user arrangements exhibit lower within-user similarity scores (Si, Si) compared to rule-based arrangements (Uj, Uj) and show higher variance in between-user similarity scores (Si, Sj) than rule-based arrangements (Ui, Uj), indicating that object arrangements in PARSEC are less rigid and more diverse than rule-based personas. The diversity and flexibility of preferences makes adaptation more challenging, motivating our decision to collect crowdsourced data.

V. Algorithm Selection

We include all the graph-based rearrangement algorithms from Table I in our evaluation. Of these algorithms, the CF [4], NeatNet [12], and TidyBot [9] techniques infer user preferences from *prior* scene context, and ConSOR adapts to preferences from *within-scene* context. TidyBot-Random is a variant of TidyBot that samples a random arrangement $\mathcal{A}^i_O$ from observed arrangements $\mathbf{A}_O$ to generate preference rules, since TidyBot requires a single example. In contrast, the CF+ [7] and APRICOT [8] models combine *prior* and *within-scene* context to place objects in partially arranged environments. APRICOT-NonInteractive, adapted from the 'Non-Interactive' baseline in the work by Wang et al. [8],



4: We propose ContextSortLM, a personalized rearrangement approach that aligns context from previously observed object arrangements with the environment's current state to ensure object placements respect user placement preferences without disrupting the environment's current arrangement.

infers a textual description of preferences from observed arrangements $\mathbf{A}_O$ at once to place objects in partially arranged environments.

ContextSortLM: We propose ContextSortLM, shown in Figure 4, for personalized rearrangement from *prior* and *within-scene* context. ContextSortLM models personalized object placement as an LLM code completion problem to infer rearrangement preferences from multiple object arrangements and place objects in partially arranged environments. Each object arrangement in $\mathbf{A}_O$ and $\mathcal{A}_P$ is rewritten as Python-style 'pick-place' commands, as shown in Figure 4A. For each prior arrangement $\mathcal{A}_O^i$ in $\mathbf{A}_O$, an LLM rule generation agent, similar to Wu et al. [9], generates a preference rule r_O^i defining where to place objects. ContextSortLM consolidates these rules into a JSON-style meta preference M_O via a separate LLM agent using the prompt in Figure 4B. A key difference in our approach over APRICOT-NonInteractive is the use of a structured preference representation that accounts for multiple valid object placements and grounds preferences in the environment surfaces S, reducing ambiguity in encoding user preferences. Objects are then placed by prompting the LLM with a code completion prompt containing M_O, the environment's current arrangement $\mathcal{A}_P$ and the list of objects and surfaces, shown in Figure 4A. In this manner, ContextSortLM aligns *prior* context and *within-scene* context to meaningfully place objects in an environment occupied with objects, while enabling zero-shot generalization to new objects and environments [4].

VI. Evaluation on PARSEC Scenarios

We utilize k-fold cross-validation to evaluate the rearrangement methods presented in Table I. Each fold in the cross-validation set includes a training set $\mathcal{D}_{train}$ and test set $\mathcal{D}_{test}$. We conduct two experiments – the KnownEnv and NovelEnvCategory experiments – to evaluate adaptation to user preferences in previously seen and unseen environments. We generate each fold by excluding the examples from one of five users per environment category and examples of all users from one of three environment categories from training respectively. On average, each fold in KnownEnv contains 2806 $\mathcal{D}_{train}$ and 701 $\mathcal{D}_{test}$ examples, and each fold in NovelEnvCategory contains 2338 $\mathcal{D}_{train}$ and 1169 $\mathcal{D}_{test}$ examples. In each fold, we separate arrangement examples of a random user from $\mathcal{D}_{train}$ as a validation set $\mathcal{D}_{val}$, resulting in approximately 85 $\mathcal{D}_{val}$ examples per fold. We performed early-stopping when training models via the average SED metric calculated on $\mathcal{D}_{val}$ examples.

[4]ContextSortLM, APRICOT-NonInteractive, and TidyBot-Random use the *gpt-4-0613* model.

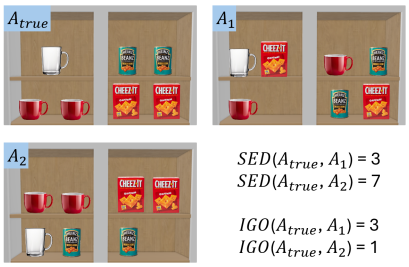


Fig. 5: SED and IGO calculated between a hypothetical user arrangement A_{true} and two possible predicted arrangements $\mathcal{A}_a$ and $\mathcal{A}_2$. Note how $\mathcal{A}_2$ has a high SED but a low IGO, since most same-category objects are grouped together as in $\mathcal{A}_{true}$ but not placed on the correct surface.

Metrics: We define two measures of similarity between the model's predicted arrangement $\mathcal{A}_G$ and the user's true arrangement $\mathcal{A}_G^*$ to assess model performance, shown in Figure 5. The *Scene Edit Distance*, or **SED**, borrowed from prior work [6], is the minimum number of objects that must be moved in $\mathcal{A}_G$ to perfectly match $\mathcal{A}_G^*$. The *Number of Incorrectly Grouped Objects*, or **IGO**, is the minimum number of objects that must be moved in $\mathcal{A}_G$ so that the same sets of objects are placed together as in $\mathcal{A}_G^*$, while ignoring the exact surface on which the objects are placed. Collectively, the SED and IGO measure deviation from the preferred surface assignment and object grouping respectively. We also compute the Placement Accuracy or PA, as defined by Wu et al. [9], which is the average number of object placements predicted for $\mathcal{X}_U$ that match the placements in $\mathcal{A}_G^*$.

Model	KnownEnv				NovelEnvCategory			
	Similar-1D	Similar-2D	Dissimilar	**Average**	Similar-1D	Similar-2D	Dissimilar	**Average**
ContextSortLM (Ours) †	**0.54**	**0.57**	**0.65**	**0.59**	**0.54**	**0.57**	**0.65**	**0.59**
APRICOT-NonInteractive †	0.50	0.50	0.56	0.53	0.50	0.50	0.56	0.53
TidyBot-Random †	0.46	0.40	0.46	0.44	0.46	0.40	0.46	0.44
ConSOR	0.36	0.41	0.35	0.37	0.32	0.38	0.27	0.31
CF	0.30	0.33	0.23	0.28	0.30	0.31	0.24	0.28
CFFM	0.23	0.22	0.23	0.23	-	-	-	-
NeatNet	0.28	0.25	0.29	0.28	-	-	-	-

TABLE II: Placement Accuracy (PA) calculated when adapting to new users in seen environments and unseen-category environments. The † indicates models not trained on any example from the dataset, meaning that they perform identically in KnownEnv and NovelEnvCategory conditions.

A. Results

We present the results of KnownEnv and NovelEnvCategory experiments. We report PA as an aggregate measure of rearrangement performance for each experiment. For the KnownEnv, we also report SED and IGO scores across different initial environment conditions to study how rearrangement performance changes over increasing *within-scene* context.

Aggregate Performance: Table II presents the PA scores for KnownEnv and NovelEnvCategory experiments. Across both experiments, ContextSortLM and APRICOT-NonInteractive achieve higher PA than TidyBot-Random and ConSOR, demonstrating the benefit of integrating multiple semantic context sources rather than relying on a single source for preference adaptation. Moreover, ContextSortLM, APRICOT-NonInteractive, and TidyBot-Random have a higher PA than other methods without any pre-training or using in-context examples from our dataset, highlighting the importance of pre-trained commonsense reasoning in adapting to new user preferences.

ContextSortLM outperforms APRICOT-NonInteractive in all environment categories, emphasizing the benefit of a structured preference representation over a textual preference description. Notably, the PA of ContextSortLM and APRICOT-NonInteractive differs the most in the `Dissimilar` category, which typically has more surfaces, resulting in fewer objects per surface. This is likely because, unlike ContextSortLM, APRICOT-NonInteractive outputs the preference description from observed arrangements $\mathbf{A}_O$ all at once, over-grouping objects and overgeneralizing user preferences.

Among non-LLM models, ConSOR considerably outperforms the bottom half of the table (CF, CFFM, and NeatNet). NeatNet and CFFM struggle due to limited training data per environment, but CF underperforms even after training across environments. CF's reliance on pairwise object similarity misses broader contextual cues essential for rearranging across different environments, resulting in lower placement accuracy. CFFM's performance also suffers due to both limited data and a lack of global semantic context, despite using both scene context sources. Incorporating global semantic context is, therefore, essential for modeling preferences across environments.

CF's PA remains relatively stable across KnownEnv and NovelEnvCategory experiments compared to ConSOR's PA, particularly in the `Dissimilar` category, suggesting that models leveraging *prior* scene context generalize better to unseen environments than those relying on *within-scene* context. However, more experiments are required.

Placement Error in Partially Arranged Environments: Figure 6 shows the placement error (SED and IGO scores) of the above models as a function of N_P, the number of objects in the environment's current arrangement $\mathcal{A}_P$. $N_P = 0$ represents an empty environment and $N_P = 12$ a densely populated one. ContextSortLM achieves the lowest mean SED in both sparsely occupied ($N_P = \{0, 4\}$) and densely occupied environments ($N_P = \{8, 12\}$), with APRICOT-NonInteractive ranking second, which further supports integrating multiple scene context sources for personalized rearrangement. However, ContextSortLM's SED scores exhibit high variance in densely occupied environment, indicating that it struggles to place objects in such scenarios. Moreover, ConSOR and TidyBot-Random have comparable mean SED scores in densely occupied environments, indicating that *within-scene* context is as effective as *prior* scene context for object placement in partially arranged environments.

Across all models, the IGO scores are lower than SED scores in sparsely occupied $\mathcal{A}_P$, suggesting that rearrangement models are better at grouping similar objects than selecting appropriate surfaces. This is likely because models leverage common object similarities encoded in external knowledge. Our finding also aligns with human tendencies to agree strongly on object similarity [15]. The SED and IGO scores converge in densely occupied environments ($N_P \geq 8$), where limited empty surfaces make object placement synonymous with grouping similar items.

VII. Evaluation with Online Raters

To assess the alignment of computational metrics with human judgment, we conducted a crowdsourced user evaluation where online raters ranked different rearrangement models based on alignment with target user's preferences. We selected four top-performing models – ContextSortLM, APRICOT-NonInteractive, TidyBot-Random, and ConSOR – and chose examples from PARSEC where all four model predictions differed. Online raters in our experiment first examined the observed arrangements, $\mathbf{A}_O$, and wrote a summary describing them. This summary serves as a quality check, allowing us to filter out raters who submit irrelevant summaries. Raters then reviewed the environment's current arrangement, objects to be placed, and the predicted object arrangements from all four models. Raters identified which predicted object arrangement perfectly matched the target user's preferences and ranked the arrangements based on

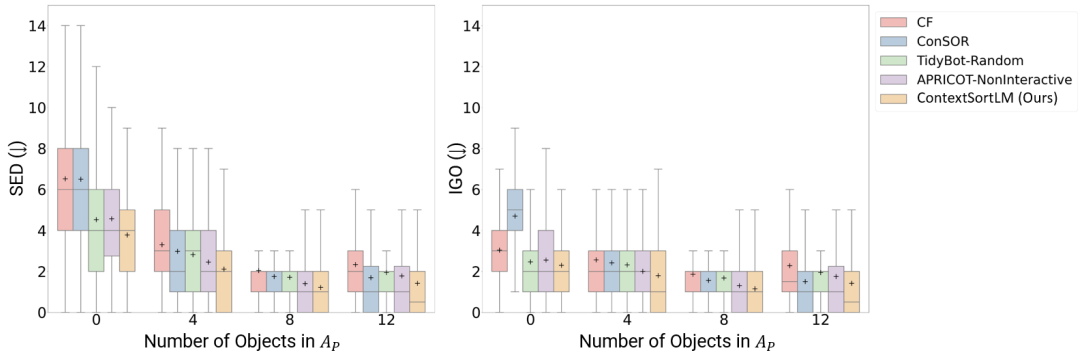


Fig. 6: Placement error metrics (SED and IGO) calculated as a function of the number of objects in the environment's current arrangement for the KnownEnv experiment. The '+' sign denotes the mean value.

alignment to user preferences. To reduce bias, we counterbalanced the order of model predictions and recruited three independent raters per example. In total, we hired 108 raters [5] to evaluate 36 object arrangement examples, spanning 14 user preferences, each with 2–3 variations of $\mathcal{A}_P$.

Evaluation Metrics: We define two metrics to analyze rater responses for each rearrangement model. Alignment Score, or s_{align}, is the percentage of raters who found the model's predicted object arrangement to perfectly align with the target user's preferences. Rank Score, or s_{rank}, is the average rank raters give to a model's predictions. A higher s_{align} indicates that the model frequently places objects to match the target user's preferences, and a lower s_{rank} signifies better alignment with the user's preferences compared to other models.

A. Results

Alignment Score: Table III presents the s_{align} metric, which measures how well models match the target user's preferences.

Our results indicate that models incorporating *prior* and *within-scene* context, ContextSortLM and APRICOT-NonInteractive, achieve higher alignment scores than TidyBot-Random and ConSOR across all three environment categories, further validating the use of multiple sources of scene context for preference adaptation. APRICOT-NonInteractive outperforms ContextSortLM in `Similar-1D` and `Similar-2D` categories but underperforms in the `Dissimilar` category, likely because APRICOT-NonInteractive tends to over-cluster similar objects. ConSOR scores higher than TidyBot-Random in the category `Similar-1D` but lags in the `Similar-2D` and `Dissimilar` categories, possibly because `Similar-1D` examples are densely occupied and offer richer semantic context benefiting ConSOR.

Many raters found no model perfectly aligned with user preferences, especially in Similar-2D examples, where 27.0% chose 'None.' `Similar-2D` environments feature identical surfaces in a 2D layout, which is challenging to represent semantically, leading to preference mismatches and misplaced objects. This underscores the challenge of modeling spatial information about previously unseen environments such as relative surface positions. Notably, some raters who selected 'None' explicitly mentioned in their summaries that the target user's preferences were unclear, which may have hindered their ability to identify a perfect match.

Rater Response	`Similar-1D` (%)	`Similar-2D` (%)	`Dissimilar` (%)
ContextSortLM (Ours)	40.5	37.8	**60.7**
APRICOT-NonInteractive	**45.2**	**43.2**	7.1
TidyBot-Random	21.4	27.0	25.0
ConSOR	26.2	24.3	28.6
None	16.7	27.0	3.6

TABLE III: Alignment scores (s_{align}), measuring how often each models matches the target user's preference. 'None' corresponds to the rater finding none of the model predictions aligning with the target user's preferences.

Rank Score: Figure 7 presents the s_{rank} metric, derived from rater-assigned model rankings and categorized by environment types. We use Friedman's one-way test followed by post-hoc Wilcoxon Signed-Rank tests [6] for statical analysis, marked in the figure. Friedman's test indicates statistically significant differences in the `Similar-1D` and `Dissimilar` category ($p < 0.001$), and post-hoc pairwise comparisons with Wilcoxon Signed-Rank test reveal that ContextSortLM has a significantly lower median rank than TidyBot-Random in the `Similar-1D` category ($p < 0.01$) and APRICOT-NonInteractive in the `Dissimilar` category ($p < 0.05$), which is consistent with previous findings.

Surprisingly, there are few statistically significant differences among models for the `Similar-1D` and `Dissimilar` categories, suggesting that users tolerate reasonable variations in object placement. The absence of any significant differences in the `Similar-2D` category is likely due to the challenges of accurately modeling spatial information in these environments, resulting in discrepancies in user preferences.

VIII. Summary and Discussion

Our evaluation results strongly support integrating *prior* and *within-scene* context for personalized rearrangement. To guide future work in better integrating the two scene context sources, we summarize key takeaways from evaluations and highlight ContextSortLM's limitations.

[5]Out of 110 raters, two were removed due to poor quality summaries.

[6]Bonferroni correction of $\alpha = 6$ was applied.

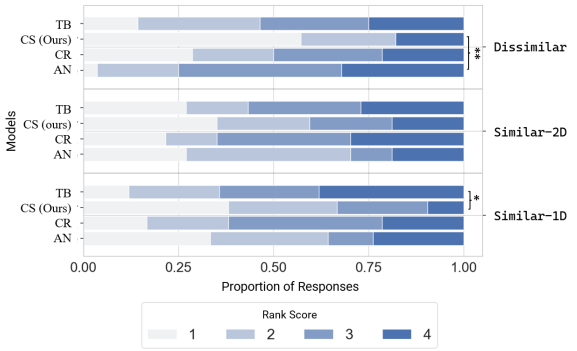


Fig. 7: Distribution of rank scores s_{rank}, derived from rater-assigned model rankings, and categorized by environment type. '0.5' on the x axis denotes the median rating. The acronyms TB, CS, CR and AN represent TidyBot-Random, ContextSortLM, ConSOR and APRICOT-NonInteractive respectively. '*' and '**' symbols indicate p-values less than 0.05 and 0.01 respectively.

Among the models integrating dual context sources, ContextSortLM outperforms APRICOT-NonInteractive when comparing computational metrics, particularly in `Dissimilar` environments, and achieves consistently high alignment and low rank scores across environment categories, highlighting the benefit of structured preference representations that explicitly account for multiple valid object placements over textual preference descriptions.

While s_{align} scores align with trends seen in computational metrics, s_{rank} scores show few significant differences among models, highlighting challenges in encoding semantic information about the environment. Raters tolerated most variations in object placements but penalized irrelevant object placements in designated easy-access surfaces, such as the cabinet's bottom shelf, or purpose-specific locations, such as the fridge's top shelf or vegetable drawer. Encoding more semantic information from the environment- such as better spatial information about surfaces and knowledge of object usage patterns- will improve preference adaptation.

Limitations of ContextSortLM ContextSortLM struggles in environments occupied with many objects due to over-reliance on its meta-preference M_O, sometimes grouping dissimilar objects. ConSOR's success in the same setting suggests a hybrid LLM/specialized-policy approach – filtering noisy *prior* scene context with LLMs and resolving conflicts with the current environment via learned policies leveraging *within-scene* context. Moreover, ContextSortLM's meta preference M_O is sensitive to noisy object placements in the observed arrangements (e.g., a coffee mug randomly placed in the fridge), and we aim to refine ContextSortLM to ignore such outlier placements in future work.

IX. Conclusion

In conclusion, we introduced PARSEC, an object rearrangement benchmark where robots adapt to user organizational preferences from scene context for object placement in partially arranged environments. PARSEC includes a novel crowdsourced dataset of 110K evaluation examples collected from 72 real users, covering 93 household objects across 15 environment instances. We also proposed ContextSortLM, an LLM-based personalized rearrangement model that places objects in partially arranged environments by adapting to user preferences from *prior* and *within-scene* context while accounting for multiple valid placements. We evaluated ContextSortLM and existing personalized rearrangement models on PARSEC and complemented these findings with a crowdsourced user evaluation of 108 online raters ranking model predictions based on alignment to user preferences. Our results highlight the importance of integrating multiple sources of scene context for personalized object placement in partially arranged environments. However, there are challenges in modeling environment semantics – such as the environment's spatial layout and utility of different environment surfaces – leading to discrepancies in inferred user preferences. Moreover, ContextSortLM outperforms other models in computational evaluations due to its structured preference representation, but struggles in densely occupied environments, emphasizing the need for improved techniques to integrate prior and within-scene context.

Finally, we note that the choice of using an LLM for our proposed approach is based on the strong performance of prior LLM-based object rearrangement models [9], [8], [2]. As LLM performance is highly dependent on the information provided in the prompt, our work provides design guidelines for future LLM-based rearrangement models.

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] B. A. Newman, P. Gupta, K. Kitani, Y. Bisk, H. Admoni, and C. Paxton, "Degustabot: Zero-shot visual preference estimation for personalized multi-object rearrangement," *arXiv preprint arXiv:2407.08876*, 2024.

[3] B. Trabucco, G. A. Sigurdsson, R. Piramuthu, G. S. Sukhatme, and R. Salakhutdinov, "A simple approach for visual room rearrangement: 3d mapping and semantic search," in *ICLR*, 2022.

[4] N. Abdo, C. Stachniss, L. Spinello, and W. Burgard, "Organizing objects by predicting user preferences through collaborative filtering," *IJRR*, 2016.

[5] R. Ramrakhya, A. Kembhavi, D. Batra, Z. Kira, K.-H. Zeng, and L. Weihs, "Seeing the unseen: Visual common sense for semantic placement," *CVPR*, 2024.

[6] K. Ramachandruni, M. Zuo, and S. Chernova, "ConSOR: A context-aware semantic object rearrangement framework for partially arranged scenes," in *IEEE IROS*, 2023.

[7] S. Brawner and M. L. Littman, "Learning user's preferred household organization via collaborative filtering methods.," in *IntRS@ RecSys*, 2016.

[8] H. Wang, N. Chin, G. Gonzalez-Pumariega, X. Sun, N. Sunkara, M. A. Pace, J. Bohg, and S. Choudhury, "APRICOT: Active preference learning and constraint-aware task planning with LLMs," in *CoRL*, 2024.

[9] J. Wu, R. Antonova, A. Kan, M. Lepert, A. Zeng, S. Song, J. Bohg, S. Rusinkiewicz, and T. Funkhouser, "Tidybot: Personalized robot assistance with large language models," *Autonomous Robots*, 2023.

[10] Y. Kant, A. Ramachandran, S. Yenamandra, I. Gilitschenski, D. Batra, A. Szot, and H. Agrawal, "Housekeep: Tidying virtual households using commonsense reasoning," in *ECCV*, 2022.

[11] R. Toris, D. Kent, and S. Chernova, "Unsupervised learning of multi-hypothesized pick-and-place task templates via crowdsourcing," in *IEEE ICRA*, 2015.

[12] I. Kapelyukh and E. Johns, "My house, my rules: Learning tidying preferences with graph neural networks," in *CoRL*, 2022.

[13] B. A. Newman, C. Paxton, K. Kitani, and H. Admoni, "Bootstrapping linear models for fast online adaptation in human-agent collaboration," *arXiv preprint arXiv:2404.10733*, 2024.

[14] 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.

[15] M. Mur, M. Meys, J. Bodurka, R. Goebel, P. A. Bandettini, and N. Kriegeskorte, "Human object-similarity judgments reflect and transcend the primate-it object representation," *Frontiers in psychology*, 2013.