LLM-enhanced Scene Graph Learning for Household Rearrangement

WENHAO LI*, National University of Defense Technology and Xiangjiang Laboratory, China
SHILONG ZOU*, National University of Defense Technology, China
ZHINAN YU*, National University of Defense Technology, China
ZHENG ZHOU, National University of Defense Technology, China
WENXUAN LI, National University of Defense Technology, China
CHENYANG ZHU†, National University of Defense Technology, China
RUIZHEN HU†, Shenzhen University, China
KAI XU†, Institute of AI for Industries, Chinese Academy of Science and National University of Defense Technology, China

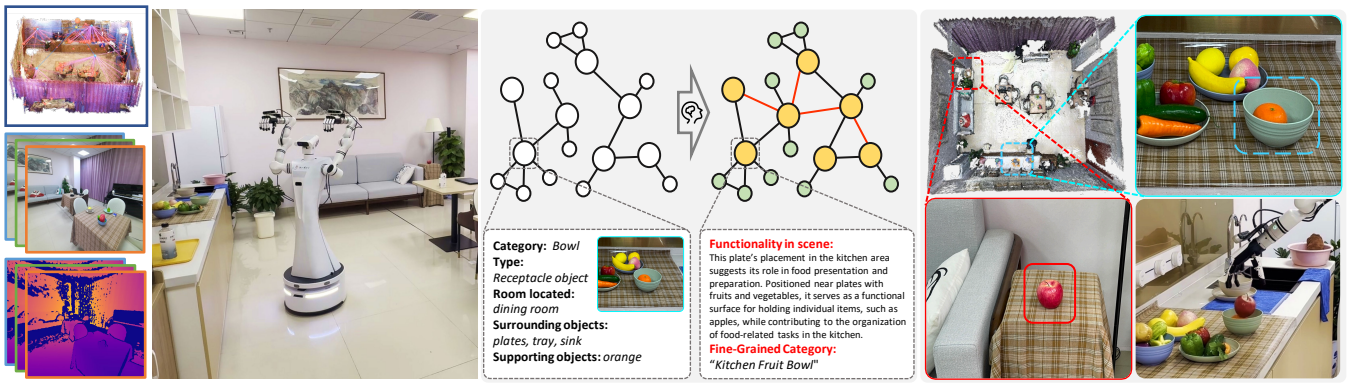

Fig. 1. Entering an indoor scene, the robot first performs an RGB-D reconstruction and scene graph extraction. It then transforms the initial scene graph into an Affordance Enhanced Graph (AEG) and plans actions for finishing the scene rearrangement task, both through prompting a multi-modal foundation model with the scene graph and selected RGB keyframes. With AEG, the robot can detect misplaced objects and determine a proper place for each of them.

The household rearrangement task involves spotting misplaced objects in a scene and accommodate them with proper places. It depends both on common-sense knowledge on the objective side and human user preference on the subjective side. In achieving such a task, we propose to mine object functionality with user preference alignment directly from the scene itself, without relying on human intervention. To do so, we work with scene graph representation and propose LLM-enhanced scene graph learning which transforms the input scene graph into an Affordance Enhanced Graph (AEG) with information-enriched nodes and newly discovered edges (relations). In AEG, the nodes corresponding to the receptacle objects are augmented with context-induced affordance which encodes what kind of carriable objects can be placed on it. New edges are discovered with newly discovered non-local relations. With AEG, we perform task planning for scene rearrangement by detecting misplaced carriables and determining a proper placement for each of them. We implement an end-to-end robot system for autonomous household rearrangement in unseen environments and test our method by implementing a tiding robot in both simulated environments and real-world scenarios, and perform evaluation on a new benchmark we build. Extensive evaluations demonstrate that our method achieves state-of-the-art performance in misplacement detection and rearrangement planning.

CCS Concepts: • **Computing methodologies** → **Artificial intelligence**; **Scene understanding**; *Computer Vision*; *Robotics*; • **Computer systems organization** → Robotics.

Additional Key Words and Phrases: Scene graph, LLM, Robotic system

ACM Reference Format:
Wenhao Li, Shilong Zou, Zhinan Yu, Zheng Zhou, Wenxuan Li, Chenyang Zhu, Ruizhen Hu, and Kai Xu. 2026. LLM-enhanced Scene Graph Learning for Household Rearrangement. 1, 1 (January 2026), 18 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Scene analysis is a long-standing problem in computer graphics and vision [Patil et al. 2024] and finds many applications ranging from

*Equal contribution
†Corresponding author

Authors' Contact Information: Wenhao Li, National University of Defense Technology and Xiangjiang Laboratory, China, davit666lwh@gmail.com; Shilong Zou, National University of Defense Technology, China; Zhinan Yu, National University of Defense Technology, China, zn_yu@nudt.edu.cn, zoushilong1024@gmail.com; Zheng Zhou, National University of Defense Technology, China, 2689638973@qq.com; Wenxuan Li, National University of Defense Technology, China, lwx_@nudt.edu.cn; Chenyang Zhu, National University of Defense Technology, China, chenyang.chandler.zhu@gmail.com; Ruizhen Hu, Shenzhen University, China, ruizhen.hu@gmail.com; Kai Xu, Institute of AI for Industries, Chinese Academy of Science and National University of Defense Technology, China, kevin.kai.xu@gmail.com.

ACM XXXX-XXXX/2026/1-ART
https://doi.org/10.1145/nnnnnnn.nnnnnnn

scene modeling/generation [Fisher et al. 2015] and VR/AR [Li et al. 2021] to various robotic applications [Zhang et al. 2023]. Among the applications, some, e.g., robot-operated household rearrangement or room tiding, require a much more in-depth analysis of the scenes than others (e.g., object goal navigation), due to the involvement of both common-sense knowledge on the objective side and human user preference on the subjective side.

When conducting a room tiding task, the agent needs to spot the misplaced objects and accommodate them in proper places [Sarch et al. 2022]. This is quite challenging, especially in an unseen environment, as it demands a nuanced understanding of each object's functionality. The determination of object functionality often requires a high-order analysis that considers a broader context [Hu et al. 2018]. Moreover, object functionality is often hinged on the house owner's preferences or intent—that is, how the homeowner chooses to use the object. For instance, a cabinet may function either as a closet or a bookshelf, depending on its location and the items placed inside. By analyzing the positioning and contextual relationships of objects in the environment, an agent can infer these preferences and determine their functionality accordingly. This enables the agent to perform personalized, user-aligned room organization in unseen environments, without human supervision, ultimately achieving more human-centered and intuitive rearrangement behavior in real-world indoor scenario.

Large Language Models (LLMs) have demonstrated remarkable capability in zero-shot planning tasks, including scene rearrangement [Kant et al. 2022; Wu et al. 2023] due to their strong common-sense reasoning capabilities. However, LLMs inherently lack scene-specific knowledge, particularly regarding the context-dependent functionality and placement of objects within a given environment. Therefore, scene grounding is crucial to ensure the practicality and relevance of the plans they generate [Rana et al. 2023].

Another important issue to consider is that LLM ignores human preference totally if the prompt is not carefully designed or tuned. In the work of TidyBot [Wu et al. 2023], this is alleviated by explicitly injecting user preference as exemplar placements via prompts and having the LLM summarize user preference out of the exemplars. Note that the exemplar placements need to be specific to the scene in consideration to make the summarization scene grounded, making the exemplar collection tedious.

Han et al. [2024] propose an iterative self-training method to align the LLM planner with user preferences, which again relies on user feedback. On the other hand, we posit that the existing layout of an indoor scene already encapsulates sufficient human preferences, which can be discerned by analyzing the scene's context, providing a natural prompt for the agent to perform rearrangement tasks.

In this work, we propose to mine object functionality with user preference alignment directly from the scene itself, without relying on human intervention. With our method, the user instruction can be as simple as "tide the house" and the agent can conduct scene rearrangement automatically based on an in-depth analysis of object functionality and proper reasoning of user preference. One example is shown in Figure 1. Based on our deep analysis of personalized functionality, it appears the user would prefer utilizing the bowl as a container for food storage, rather than as a dining utensil. While this may diverge from the conventional use of a bowl, it aligns with the user's unique preferences and contributes to a more personalized and comfortable living space.

To achieve that, we propose LLM-enhanced in-depth scene analysis to extract affordance for each object in the scene, reflecting simultaneously common-sense knowledge and personalized preference, both with scene grounding. To do so, we adopt 3D scene graphs as an LLM-friendly representation for scene analysis. Starting with a vanilla scene graph that encodes basic object categories and local spatial relations, together with its corresponding RGB sequence (or an RGB-D sequence to construct the 3D scene graph when it is unavailable), we harness LLM to enhance it into an Affordance Enhanced Graph (AEG) with information-enriched nodes (objects) and newly discovered edges (relations). In particular, the multimodal foundation model is provided with the initial scene graph (in XML format) together with keyframes selected from the RGB sequence, enabling it to leverage both graph-internal context and visual observations to infer context-induced affordances and discover new semantic relations.

With AEG, we perform scene rearrangement in two steps. The first step is misplacement detection. For each carriable object, we input its context-induced affordance, along with the affordances of its associated receptacle objects and the task instruction, into the LLM. LLM then outputs for each carriable object a probability of misplacement. In the second step, we choose the top few most probably misplaced carriables and query LLM for potential receptacles for correct placement. A straightforward method for receptacle query is to include all candidate receptacles in a single prompt. However, this results in excessively long prompts, which can degrade accuracy and increase the risk of hallucinations. Alternatively, having the LLM individually score and re-rank all possible object–receptacle pairs can improve accuracy, but is highly computationally expensive, as it requires numerous token-intensive evaluations for each placement decision [Li et al. 2024]. This makes it difficult to deploy such methods in real-world scenarios.

We, instead, opt to pre-compile task-related affordance information—i.e., affordance descriptions contextualized with the rearrangement task and misplaced objects—for all receptacles into an external memory, following the paradigm of Retrieval-Augmented Generation (RAG). During inference, only the most relevant entries are retrieved and included in the LLM prompt for placement decision-making, significantly reducing token usage while maintaining high decision quality. This optimization reduces the number of LLM inference calls from $O(mn)$ to $O(m + n)$, where m is the number of misplaced objects and n is the number of receptacles in the scene.

We contribute a new benchmark for evaluating the performance of preference-aligned scene rearrangement through annotating the Habitat Synthetic Scenes Dataset (HSSD 200) [Khanna et al. 2023] and the Arm PointNav Dataset (APND) [Ehsani et al. 2021]. In our benchmark, each carriable object is annotated with multiple valid receptacles, accompanied by a preference ranking that reflects human common sense and scene-specific expectations. This multi-ground-truth annotation better captures the diversity and ambiguity of real-world object placements.

We also implement a robotic system for end-to-end indoor scene rearrangement. Entering an unseen indoor scene, the robot begins with active exploration to acquire an RGB-D reconstruction of the

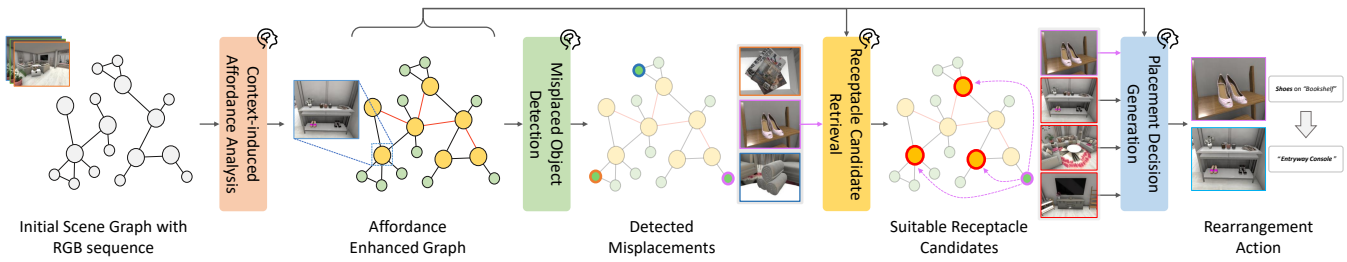

Fig. 2. Given a scene graph (SG) with RGB sequence, we utilize Large Language Model and Visual Language Model to perform context-induced affordance analysis for all objects in the scene. These affordances are incorporated into the SG, updating the nodes and edges to construct an Affordance Enhanced Graph (AEG). We then evaluate the appropriateness of the current placements of all carriable objects in the AEG based on the affordance information of the objects and their receptacles, identifying misplaced items. For each misplaced carriable, we rate the suitability of each receptacle in the AEG as a placement target. The top k suitable receptacles are selected as candidates, and their affordances are retrieved as prompts to the LLM to generate the placement decision.

scene and constructs an initial vanilla scene graph. This graph is then transformed into an Affordance Enhanced Graph (AEG), which, together with selected RGB keyframes, is used to prompt a multimodal foundation model for fully autonomous rearrangement planning and execution. We test our method in both real-world settings and in the Habitat 3.0 simulator [Puig et al. 2023].

Our main contributions include:

- We introduce LLM-enhanced scene graph learning for deep mining of object functionality with user preference alignment, which results in an affordance enhanced graph (AEG) encoding context-induced affordances of receptacles.
- Based on AEG, we propose LLM-assisted misplacement detection and correct placement determination, with carefully designed mechanisms for improved token efficiency.
- We implement a real-robot planning system based on AEG for end-to-end autonomous indoor rearrangement in unseen scenes, and test our method in real-world scenarios.
- We provide the first benchmark of house-level preference-aligned scene rearrangement by annotating a scene dataset with multiple ground-truth following a preference ranking that aligns with human intuition. With the benchmark, we test our method and achieve state-of-the-art performance.

2 Related Works

Scene understanding and affordance analysis. Scene understanding is an important area of research that can be broadly categorized into perception tasks [Georgakis et al. 2017; Silberman et al. 2012; Zhang et al. 2020] and application tasks [Achlioptas et al. 2020; Azuma et al. 2022; Duan et al. 2022]. As the field of embodied AI develops, scene understanding research has become more focused on understanding how scenes support the behavior of embodied agents such as human avatars and robots, i.e. affordance [Gibson 1977].

Affordance prediction has been approached at both the object level and the scene level. Object-level methods focus on understanding how humans use items [Deng et al. 2021; Fang et al. 2018; Koppula and Saxena 2014; Nagarajan et al. 2019, 2020], while scene-level research emphasizes how to naturally place a human avatar within a scene [Grabner et al. 2011; Gupta et al. 2011; Kulal et al. 2023; Li et al. 2019; Savva et al. 2014, 2016; Wang et al. 2024; Zhu et al. 2016]. Different from these works which analyze affordance from a human interaction aspect, some studies focus on the analysis of object-object affordance [LI et al. 2022; Mo et al. 2022; Tang et al. 2021]. Their methods predict the physical interaction between objects, such as placement and stacking. [LI et al. 2022; Tang et al. 2021] extract potential object interactions within a given scene. However, all these object-object affordance works do not consider the "scene-grounding affordance" of the objects, which requires an in-depth analysis of factors including the object's geometry, pose, and the scene context, such as its position in the scene and the objects interacting with it.

Foundation models for scene understanding. Visual language models [Radford et al. 2021] and large language models [Achiam et al. 2023] have been recently used in scene understanding tasks [Hong et al. 2023; Jatavallabhula et al. 2023; Wang et al. 2023]. Some works [Li et al. 2023; Nguyen et al. 2023; Van Vo et al. 2023; Zhang et al. 2024] use or distill the rich commonsense provided by foundation models to identify affordance for a sole object. [Ni et al. 2023; Rajvanshi et al. 2023; Rana et al. 2023] use the scene graph to ground LLMs for navigation task. However, they just use the scene graph as a map without extracting object interaction information hiding in the scene graph. While in our work, we focus on using scene graph to ground LLMs in analyzing the object affordances, the results are then sent back to enhance the scene graph for deeper analysis.

Rearrangement. Rearrangement is a special embodied AI task to test the robots' intelligence [Batra et al. 2020]. In a given initial state of the scene, robots are tasked with manipulating the poses of objects to align with desired goal states. These goal states can be represented through various means such as object pose transformations [Liu et al. 2021], language instructions [Liu et al. 2023, 2022], or visual images [Huang et al. 2023; Weihs et al. 2021]. It is worth noting that in certain cases like household tidying, robots may need to rearrange objects based on the original scene and their experiences. This necessitates the robot to have human-like common sense and the ability to decipher user preferences embedded within the original scenarios. Previous works in household rearrangement rearrange objects without considering the fine-grain affordances constrained by the scene [Kant et al. 2022; Sarch et al. 2022]. In our

method, we analyze the object affordance with scene graph and LLMs, which enable us rearrange the households while follow the original functionality of the object as much as possible.

3 Method

Figure 2 presents an overview of our approach. Starting with a 3D scene, abstracted into a scene graph and its corresponding RGB sequence our method initiates by harnessing the contextual details of each node. We select representative keyframes for all nodes from RGBs and extract nodes/edges information to augment the graph using the Large Language Model and Vision-Language Model, infusing it with induced affordances and uncovering new semantic connections between objects. The resulting Affordance Enhanced Graph (AEG) enriches the initial scene graph with context-induced object functionalities and functional semantic relationships, providing a more expressive representation for household rearrangement.

Adopting the objective of "tidying the house," the affordance-enriched graph serves as the foundation for identifying objects that are out of place. To determine the most suitable location for each misplaced object, our system initially extracts a set of potential receptacle candidates from the graph. These candidates are then meticulously evaluated and compared to arrive at the optimal placement decision. It is important to highlight that all the pivotal steps of our methodology are executed in a zero-shot learning context, leveraging the advanced capabilities of the LLM and VLM.

Furthermore, building on the above methodology, we implement a real-robot system capable of performing end-to-end household rearrangement in previously unseen environments. When an initial scene graph is unavailable, the robot autonomously explores the environment to acquire aligned RGB-D observations and construct the initial scene graph from scratch, after which it executes grasping and placement actions guided by the enhanced scene graph and LLM-generated plans.

3.1 Context-induced Affordance Analysis

In this section, we describe how the initial scene graph (SG) is enhanced by leveraging a Large Language Model (LLM) to analyze contextual information, as illustrated in Figure 3. We extract three types of context from the SG and jointly feed them into the LLM to perform affordance analysis.

For each object, we first extract the intrinsic information stored in its SG node, along with its spatial relationships to neighboring objects represented by connecting edges. We refer to this as the object's textual context. Next, we select a keyframe from the RGB sequence that best captures the object and its surrounding details, and use the Vision-Language Model (VLM) to derive the object's visual context from this keyframe. Finally, we construct a hierarchical structure over the SG and employ the LLM to aggregate information in a bottom-up manner, identifying spatially distant yet functionally related objects. The aggregated information and newly discovered relationships are defined as the global context.

With these three contexts and a designed prompt, the LLM infers each object's context-induced affordances. The extracted contexts and inferred affordances are stored in the corresponding SG nodes, and newly discovered functional relationships are added as semantic edges, thereby enriching the initial SG. In this way, the affordance-enhanced graph augments the original representation by enhancing object nodes with contextual functionality information and extending the original local spatial structure (e.g., *on*, *near*, *support*) with higher-level functional connections inferred from context.

Initial scene graph. To simplify the problem setting, we assume that a vanilla scene graph S has already been constructed from an RGB-D scanning sequence of the given scene. In this scene graph, each node represents an object instance, and an edge connects two nodes if the corresponding objects are spatially proximate.

Each node encodes several attributes of the object instance, including the *instance ID*, *category label*, *room label* (which specifies the room in which the object is located), and *3D positions*, which represent the coordinates of the eight corners of the object's 3D bounding box. Edges capture the spatial relationships between pairs of nearby objects. These relationships are categorized into three types—*near*, *on*, and *support*—and are stored as edge attributes. We also propose a framework that enables a robotic agent to actively explore an unseen environment, perform RGB-D scanning and scene graph construction. Detailed instructions are provided in section 3.4.

Textual context extraction. Textual context refers to the intrinsic and neighboring information encoded in the initial scene graph S. For each object, we extract its category label and characterize its role in the rearrangement task. Following the interaction-based object taxonomy in OVMM [Yenamandra et al. 2023], we categorize objects into three *rearrangement types*: *carriable*, *receptacle*, and *other*, reflecting whether an object can be manipulated, can support the placement of other items, or should remain static while providing contextual information. *Carriable* objects are those that can be manipulated and transported by agents, and thus are candidates for misplacement detection and rearrangement. *Receptacles* are objects (typically flat, horizontal household surfaces) that can support the placement of carriables and therefore act as destination targets. *Other* objects are neither carriables nor receptacles; they are not directly rearranged in our setting but provide valuable context (e.g., surrounding items and local layout) for affordance inference. In our task setting, only *carriable* objects are eligible for rearrangement, *receptacle* objects define the candidate placement targets, and all types of objects contribute contextual information that supports affordance reasoning.

For each object in the scene graph S,we additionally extract its room label and all other textual information related to its neighboring objects, organizing the information using a predefined template. For example, as shown in Figure 3(b), we classify "Table-1" as a *receptacle* and gather all relevant information from its node as well as connected edges to construct its textual context. This context is then incorporated into the prompt for affordance analysis.

Visual context analysis. Relying solely on information from the initial scene graph often falls short in capturing fine-grained details of objects, such as their appearance, size, and precise position. To address this limitation, we assign a *key frame* that best captures the visual information of each object to its corresponding node, and introduce *visual context* to enrich object descriptions and support the LLM in reasoning about object functionality.

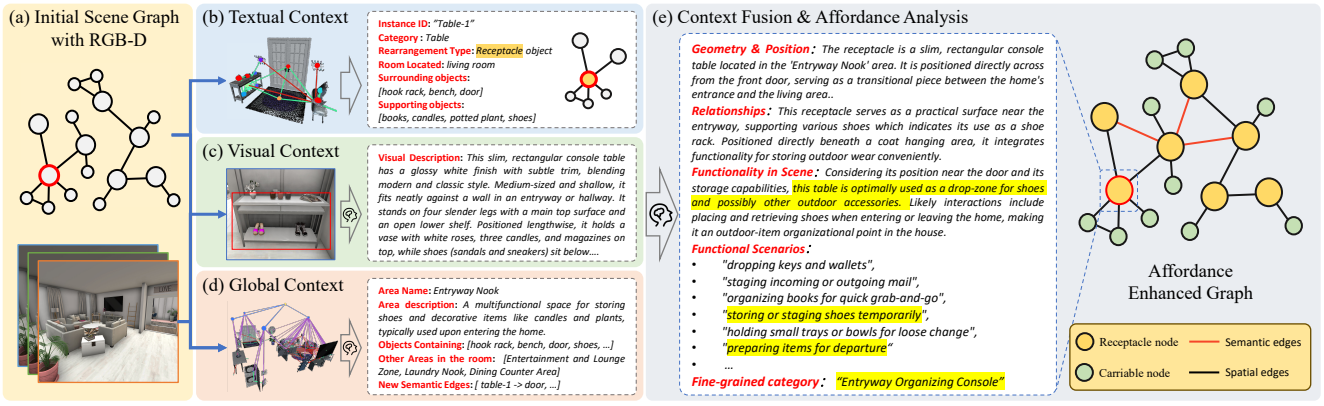

Fig. 3. Context-induced affordance analysis for objects in the scene graph. (a) Starting with an initial scene graph constructed from an RGB-D sequence, we perform the following steps for context-induced affordance analysis: (b) we extract contextual information from each object node and its neighboring nodes within the graph to get textual context; (c) we select key frames that best represent each object and use a Vision-Language Model to generate visual context; (d) we build an object-area-room hierarchy, assign key frames to each area, and use a VLM to generate area-level descriptions and infer semantic edges between spatially distant but functionally related objects as global context; (e) we integrate the textual, visual, and global contexts into a unified prompt for the LLM, which performs affordance reasoning and updates the scene graph to Affordances enhanced graph(AEG).

To incorporate visual context, we associate each object node with a representative RGB frame selected from the input RGB-D scanning sequence. Specifically, for each object instance, we compute the number of its projected pixels across all RGB-D frames, resulting in an object-to-image pixel matrix of size $N \times M$, where N is the number of object instances and M is the number of frames. To identify the most informative frame as *key frame*, we consider the projected pixel counts of both the target object and its neighboring objects. For each frame, we sum the pixel counts of the target object and all its neighbors in initial scene graph, and select the frame with the highest total as the key frame. This frame captures the richest visual information about the object and its surrounding context.

Once key frames are selected, we generate visual context by feeding each object's category label and corresponding key frame into a Vision-Language Model (VLM). The VLM analyzes the image and produces a description of the object based on its visual appearance. This description is stored as the object's visual context. For example, in Figure 3(c), the key frame of "Table-1" is selected from the RGB sequence, the object is highlighted with a red bounding box, and the frame is fed into the VLM to obtain its visual description.

Global context analysis. While the initial scene graph S captures local spatial relationships, it often overlooks functionally important but spatially distant object pairs. For example, a sofa and a television typically share a strong functional relationship (e.g., people sit on the sofa to watch TV), yet they are often far apart. As a result, they may neither be connected in S nor co-appear in the same key frame, leading the LLM to miss critical contextual cues during affordance reasoning. To address this limitation, we incorporate *global context* by constructing a hierarchical *object-area-room* structure based on S. This hierarchy enables room-level aggregation of functionally relevant information by enriching S with area-level semantic summaries, which allows the model to reason about functionally related but spatially distant objects in the room (shown in Figure 4).

We introduce an intermediate area layer to S in which objects are first grouped into areas and then aggregated into rooms. To form areas, we compute a distance matrix using 3D bounding box corners of non-carriable objects (carriables are too small to inform the scene layout). Agglomerative clustering [Ackermann et al. 2014] is applied, with a silhouette score [Shahapure and Nicholas 2020] sweep to determine the optimal number of clusters k, and objects are then grouped accordingly. Clusters with fewer than two objects are merged with the nearest larger clusters based on spatial proximity. and excluded carriable objects are later re-associated based on their supporting objects. To represent each area, we also assign a key frame analogous to the object-level key frames. The frame that contains the greatest number of objects whose projected pixels exceed a threshold is designated as the key frame for that area.

For each area, we first gather the textual context from S for all contained objects, along with the area's key frame, and input them into a VLM to generate the *area name*, *area description*. Subsequently, we aggregate information from all areas within the room and perform inference over the objects using an LLM to derive semantic edges. Using a designed prompt, VLM is guided to identify functionally related but spatially distant object pairs and to add corresponding semantic edges. By integrating these semantic edges and area-level descriptions, we enrich the scene graph with global context, enabling more comprehensive affordance reasoning. In our formulation, we define the global context for each object as the semantic edges that are relevant to the object, together with the area-level descriptions of all areas within the same room, which collectively provide a compact summary of the room layout and functionality. We then combine this global context with the object's textual and visual contexts and provide them to the LLM to perform context-induced affordance analysis.

Context Fusion and Affordance Analysis. In the final stage, we use a Large Language Model (LLM) to fuse the previously extracted

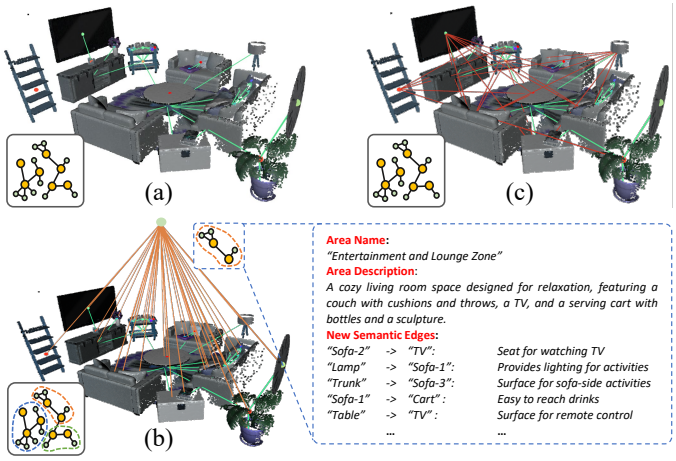

Fig. 4. Global context analysis for the scene graph. (a) In the initial scene graph, edges connect spatially proximate objects and represent only basic spatial relationships. (b) We construct an object-area-room hierarchy and aggregate information from all objects within each area. Using an LLM, we summarize the functionality of each area, generating descriptive annotations and extracting distant but meaningful relationships as new semantic edges. (c) These semantic edges are added between functionally related objects within the same room, and affordance attributes are updated in the corresponding nodes.

contexts and perform context-induced affordance analysis. For each object in the scene graph S, we construct a prompt using its stored *textual context* and *visual context* from the object's node as well as *global context*, which includes all area-level descriptions within a room and object-related semantic edges derived from the hierarchical scene structure. The LLM processes this prompt and outputs a structured affordance analysis, covering five aspects: *Geometry & Position*, *Relationships*, *Functionality*, *Functional Scenarios*, and *Fine-grained Category*.

Figure 3(e) illustrates the affordance analysis result for an example receptacle. In this example, the LLM first analyzes the geometric features of "table-1" using information from its visual context. It then integrates this with the contextual data from the textual and global context to infer that the table is positioned against the wall near the entrance of the living room. Using this position information and the table's supporting relationships in the scene, the LLM deduces that the table is likely intended for placing outdoor items. Within the *Functionality* and *Functional Scenario* modules, the LLM further reasons that the table is well-suited to store items that are usually found near entrances, such as shoes, keys, and other personal belongings for departure. Based on this comprehensive analysis, the LLM assigns a new fine-grained category to the object: *Entryway organizing console*. This approach to context-induced affordance reasoning leverages local spatial context, fine-grained visual cues, and distant semantic relationships, resulting in a deeper and more nuanced understanding of object functionality within indoor environments.

We incorporate the results of the affordance analysis into the corresponding object nodes, enriching the scene graph with updated attributes, key frames, hierarchical structure, and newly inferred

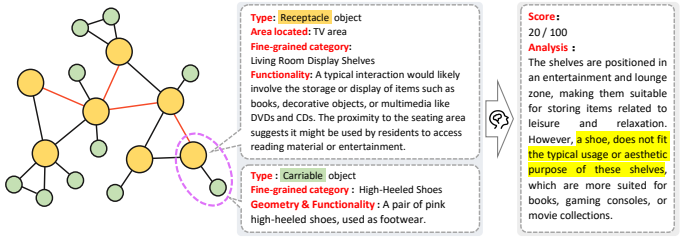

Fig. 5. An example of placement scoring with the LLM. We set a fixed standard in the prompt for the LLM to refer to when rating: 100 for perfect placement, 0 for wrong placement, and 50 for placements difficult to judge.

semantic edges. This enhanced representation is referred to as the *Affordance Enhanced Graph (AEG)*. This AEG serves as a foundational structure for downstream household tasks, such asmisplaced object detection and rearrangement planning.

3.2 Misplaced Object Detection

Since our task is to "tide the house", we are looking for carriables that are currently have an unreasonable placement and need to be rearranged to complete the task. To achieve this, we introduced an LLM-based placement scorer to rate the suitability of placements based on the affordance and description of carriable-receptacle pairs. With this placement scorer, we can determine if a carriable object is appropriately placed currently and find all the misplaced carriables.

Specifically, due to the token limitation of the LLM, we cannot input all the carriable objects at once for scoring, so we score them one by one. For each carriable object, we collect its textual description, the context-induced affordance of the receptacle where it is currently located, and our task instruction, which is"tide the house." With the collected information above, as well as a well-designed prompt, LLM will analyze the current placement of the given carriable object to determine the suitability of the placement and assign a 0˜100 score to this relevance. Figure 5 shows an example of the scoring process. We set 50 points as the threshold, and all those carriable objects with scores no more than 50 are considered to be misplaced and should be rearranged. These carriables are then reorganized in accordance with their respective scores, arranged in ascending order, to facilitate the planning of subsequent rearrangements.

3.3 Object Rearrangement Planning

For each misplaced object, our goal is to identify the most suitable receptacle in the scene as its placement target, leveraging the context-induced affordance information stored in the Affordance Enhanced Graph (AEG). To allow LLM to find the most suitable placement, it is necessary to consider all potentially relevant receptacles for comparison. However, directly inputting affordance information for all receptacles into the LLM will overwhelm the model with excessive and often irrelevant information, which impairs its reasoning ability and increases the risk of hallucinations. One potential solution is to have the LLM re-rank all placement options by using the LLM placement scorer individually evaluating the affordance of each object-receptacle pair [Li et al. 2024]. While this approach avoids input overload, it is computationally expensive

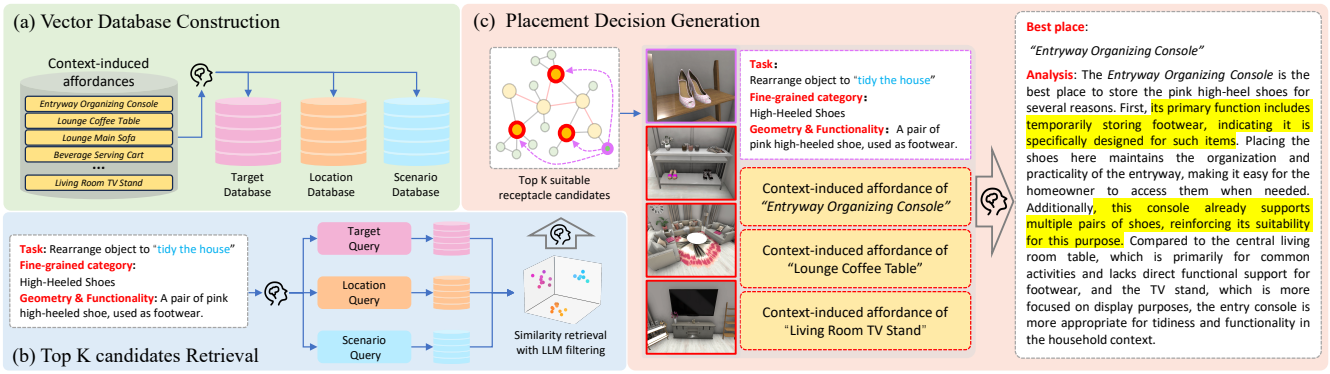

Fig. 6. We adopt a multi-query retrieval-augmented approach for placement decision generation. (a) We first use LLM to extract keywords from the context-induced affordance in AEG to construct three specialized affordance databases: the *Target Database*, *Location Database*, and *Scenario Database*. (b) For each carriable object to be rearranged, we input the original placement query into the LLM, which reformulates it into three query types—each corresponding to one of the affordance databases. These queries are used to retrieve candidate receptacles by computing semantic similarity with the database entries. An additional LLM-based filtering step is applied to eliminate low-relevance candidates from the retrieval results. (c) The top-k unfiltered receptacles with the highest aggregated similarity scores are selected as the final placement candidates.

due to the large number of token-intensive evaluations required for each placement decision.

To address this, we adopt a Retrieval-Augmented Generation (RAG) approach [Gao et al. 2023], which enhances LLM performance by retrieving only the most task-relevant text chunks from a structured database. This strategy not only reduces token consumption but also mitigates hallucinations by narrowing the input space. In our setting, this retrieval is implemented through semantic similarity search. However, applying semantic retrieval directly to household rearrangement tasks poses a unique challenge. Carriable objects and their ideal receptacles often lack strong semantic similarity (e.g., a keychain vs. an entryway console), making embedding-based retrieval unreliable. Inspired by the RAG-Fusion method [Rackauckas 2024], we address this issue by decomposing the original placement query (i.e., "Where should this object be placed to tide the house?") into multiple sub-questions to align with the context of affordance information encoded in the Affordance Enhanced Graph (AEG). These include:

- *What types of receptacles are suitable for this object as placement targets?*
- *Where (in which room or area) are such targets typically located?*
- *What are the underlying functional scenarios behind placing this object on those specific targets for the house tidying task?*

Each sub-question corresponds to a specific searchable dimension of the AEG. Using a carefully designed prompt, the LLM answers these sub-questions to generate multiple structured queries. These queries are then used to search the associated specialized vector databases, which are constructed by leveraging the LLM to extract question-relevant contextual dimensions (e.g., object category, room and area location, and functional scenarios in the scene) from the affordance descriptions of all receptacles in the AEG.

From these databases, we retrieve the top-k task-relevant receptacles as candidate placements. To further refine the results, we introduce an LLM-based filtering step that removes low-relevance candidates. This combined strategy of multi-query retrieval and LLM-based filtering ensures that only the most relevant receptacles are retained. The remaining candidates, each enriched with affordance context, are then passed to the LLM for final placement decision-making. This pipeline enables the LLM to infer accurate and context-aware placements for carriable objects, resulting in high-quality rearrangement plans that are both token-efficient and reliable. The overall process is illustrated in Figure 6.

Vector Database Construction. After constructing the Affordance-Enhanced Graph (AEG), we build three specialized databases to support placement retrieval, each corresponding to one of the sub-questions introduced earlier. These databases are defined as follows:

- **Target Database** DB_T**:** Captures the initial category and inferred fine-grained category of each receptacle.
- **Location Database** DB_L**:** Stores the spatial information of each receptacle, including the room and area it belongs to.
- **Scenario Database** DB_S**:** Encodes the intended functional scenario of each receptacle within the scene.

To populate the databases, each receptacle's context-induced affordance description is processed by an LLM using a structured prompt to extract three key types of information: (1) its initial and fine-grained categories for DB_T, (2) its room and area labels for DB_L, and (3) its functional scenario within the scene for DB_S. The extracted content is vectorized and stored as semantic indices, enabling efficient and context-aware retrieval for placement planning.

Receptacle candidate retrieval. For each misplaced object, we apply a similar strategy to generate queries using an LLM. Based on the three sub-questions introduced, the model produces multiple queries per question, which are used to search the corresponding databases:

- **Target Queries** Q_T**:** What category of receptacles is most suitable for this object?

- **Location Queries** Q_T**:** Where (in terms of room or area) is this object most appropriately placed?
- **Scenario Queries** Q_S**:** What unique functional scenario is enabled when this object is placed correctly for house tidying task?

These queries are vectorized and compared against entries in their respective databases using cosine similarity as a retrieval score. For instance, given a carriable object obj and a receptacle rec, the retrieval score for the target database is calculated as follows:

$$Score_T(obj, rec) = \max_{\mathbf{q}_T^i \in Q_T(\text{obj})} \left(\frac{\mathbf{q}_T^i \cdot \mathbf{Emb}_T^{\text{rec}}}{\|\mathbf{q}_T^i\| \, \|\mathbf{Emb}_T^{\text{rec}}\|} \right)$$

where q_T^i is one of the multiple vectorized queries from generated target query $Q_T(obj)$, and emb_T^{rec} is the vectorized embedding of context data about rec's fine-grained category from target database DB_T. For each receptacle candidate, we compute the retrieval scores across the three query types and sum them to get a final retrieval score. All candidate receptacles are ranked based on their final retrieval scores. From this ranking, we select the top-k candidates and input their fine-grained categories into the LLM for a lightweight filtering stage, where unsuitable options are identified and removed. Any filtered-out candidates are replenished from the original ranked list to maintain a complete top-k set. This final candidate set, enriched with context-induced affordance information, is then used as retrieval input for generating the placement decision.

Placement decision generation. We introduce an LLM-based placement decision generator to select the best receptacle among the proposed candidates and generate a rearrangement plan. With the top-rated receptacles, we feed their context-induced affordance as well as the initial query into the decision generator. The LLM selects the best placement target, outputting the placement decision with a paragraph of analysis. With the generated placement target, we can pick up all the misplaced carriable objects and put them in their target locations, completing the rearrangement process.

3.4 End-to-End Household Rearrangement

Many existing simulation benchmarks (e.g., Habitat [Puig et al. 2023]) have significantly facilitated research on room rearrangement by simplifying the problem setting. However, deploying our algorithms on real robots to perform rearrangement in previously unseen, real-world environments presents substantial challenges. This is because simulated benchmarks often rely on several key assumptions: operating in a fully observable environment, having access to an initial scene graph and RGB-D sequence describing the scene, and allowing the robot to navigate to any object and perform grasping and placement using a simplified suction mechanism.

In real-world settings, these assumptions do not hold, necessitating direct solutions to these challenges. To address them, we designed a fully integrated robotic system capable of active scene exploration, scene graph construction, and autonomous rearrangement task execution, enabling the robot to apply our proposed method and achieve end-to-end household rearrangement effectively in previously unseen, real-world environments.

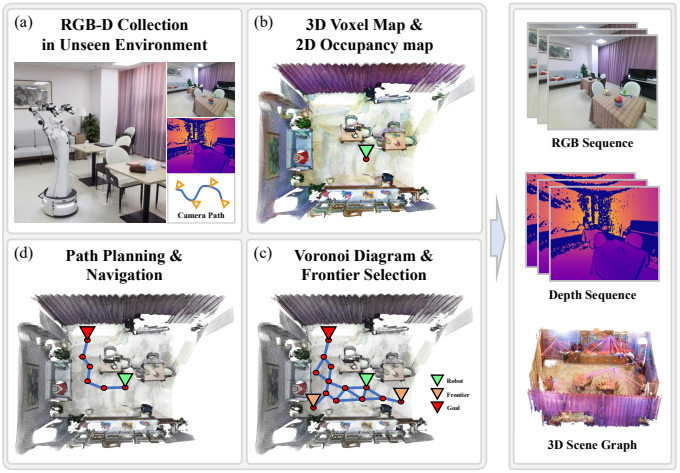

Fig. 7. Active Exploration in an Unseen 3D Scene. (a) The robot captures a sequence of RGB-D images, which are used to calculate point clouds of the environment. (b) Point clouds are voxelized into a voxel map and then projected into a 2D occupancy grid for exploration guidance. (c) A Generalized Voronoi Diagram (GVD) is constructed to skeletonize the traversable regions in the 2D space. Frontiers are generated, and the farthest frontier is selected as the next goal. (d) A* is used to compute the path, and the robot navigates to the goal, collecting additional RGB-D sequences along the way. The process from (a) to (d) is repeated until no frontiers remain, at which point the robot has completed the RGB-D scan of the entire scene.

Active Exploration. Achieving complete reconstruction of an unknown environment using only the robot's onboard RGB-D camera is a challenging problem. The primary difficulty lies in enabling the robot to explore without a prior map while ensuring sufficient coverage of the entire scene. This requires a path planning mechanism that simultaneously accounts for the geometric structure of the environment and the efficiency of exploration. Following CogNav [Cao et al. 2024], we adopt a Generalized Voronoi Diagram (GVD)-based approach [Okabe et al. 1994] to guide the robot's exploration path planning during the reconstruction process.

As shown in Figure 7, when the robot is placed in an unseen environment, it first performs a 360-degree scan to obtain a sequence of RGB-D images. These RGB-D sequences are converted into point clouds, which are then voxelized to produce a 3D voxel map representing the observed environment. From this map, we extract all voxels within a vertical height range and project them onto a 2D occupancy map, which is used as the robot's navigation map. Based on the 2D map, we construct a GVD to skeletonize the traversable regions in the 2D space, and merge spatially proximate nodes into a single node. The simplified GVD reflects the structure of the explored space and is used to guide exploration. We then select all nodes with only one edge in GVD as potential frontiers and choose the farthest one as the short-term goal. Using A* [Berkhemer et al. 2015] on the 2D map, the robot navigates to the goal.

During navigation, the robot incrementally acquires new RGB-D sequences. These new sequences are converted into point clouds, voxelized, and fused with the accumulated voxel map. As the robot explores, the GVD and frontiers are continuously updated. This process continues until no new frontiers remain. then the robot has

completed its exploration, obtaining a full RGB-D sequence and 3D point cloud of the entire scene.

Scene Graph Construction. With the RGB-D scan sequence and the corresponding 3D point cloud obtained from the robot's active exploration, we employ the OVIR-3D method [Lu et al. 2023] to detect all objects' 3D bounding boxes and construct the initial scene graph from scratch. Specifically, we first apply instance segmentation to each RGB image in the sequence, generating multiple instance masks. For each segmented region, we extract CLIP features [Radford et al. 2021]. These instance masks and features are then back-projected onto the corresponding 3D point clouds, resulting in 3D candidate regions.

Next, similar 3D candidate regions are merged by comparing their Intersection over Union (IoU) and feature similarity, producing a set of distinct instances in the scene, each with a corresponding 3D bounding box in the point cloud. These instances are treated as nodes, and edges are created between them based on the spatial relationships of their bounding boxes. This process yields a 3D initial scene graph of the entire environment, along with the associated RGB-D frames, which are subsequently used for affordance enhancement by the LLM.

Rearrangement Task Execution. For the rearrangement task, once all misplaced objects and their corresponding placement targets have been identified using the AEG, the robot executes the task through iterative grasping and placement operations.

Based on existing methods such as AnyGrasp [Fang et al. 2023], robots can already obtain high-quality 3D grasp poses for real-world manipulation tasks. However, these approaches are highly dependent on the quality of captured images, which becomes particularly challenging in mobile manipulation scenarios. Poorly chosen navigation targets can significantly degrade both image capture quality and subsequent manipulation performance, often preventing the robot from acquiring accurate grasping poses. Even when reliable grasp poses are successfully generated, the robot may still fail to execute the grasp due to inaccessible target positions or potential collisions during manipulation.

However, since the AEG stores the 3D bounding box of each object in the scene, we can leverage this geometric information together with the camera's intrinsic parameters to generate candidate camera viewpoints above the target object using a simple heuristic. These viewpoint candidates are designed to ensure sufficient image coverage and a feasible viewing distance under the robot's workspace constraints. Based on these candidates, we further incorporate the robot's kinematic structure and the 2D occupancy map to compute feasible robot base poses and navigation goals, enabling the robot to reliably move to the desired observation position. After navigation and image capture, we apply DINO [Ren et al. 2024] for object detection and AnyGrasp to generate the object's grasp pose. To address feasibility issues, we implement a twin-inference framework, which performs reachability and collision checks in simulation ahead of execution. This framework filters and selects the most feasible grasp pose, ensuring robust real-world grasp execution.

Similarly, when an object needs to be rearranged, the robot computes the navigation target in the same manner to reach the placement location. It then captures an image, calculates the optimal placement position based on the point cloud and RGB image, and executes the placement action using its robotic arm to position the object at the designated location. Once all actions are completed, the LLM updates the affordances of the corresponding nodes in the AEG accordingly.

The implementation details for active exploration, SG construction, and task execution are provided in supplementary materials.

Table 1. Comparison to Tidybot on its benchmark by sorting criteria. We calculated the average success rate of different sorting criteria by weighting the success rate with the number of objects within that criteria.

Method	Category	Attribute	Function	Subcategory	Multiple	Average
Tidybot	90.07%	87.91%	87.04%	**93.64%**	91.66%	90.10%
Ours	**91.42%**	**88.76%**	**88.18%**	91.26%	**94.82%**	**91.22%**

4 Experiment and Results

4.1 Experiment Setup

We evaluate our method on two benchmarks: one introduced by TidyBot [Wu et al. 2023], and another proposed in this work—a context-oriented benchmark constructed and annotated using the Habitat Synthetic Scenes Dataset (HSSD 200) [Khanna et al. 2023] and the Arm PointNav Dataset (APND) [Ehsani et al. 2021].

TidyBot Benchmark. We assess our method on the object rearrangement planning task using the benchmark from TidyBot [Wu et al. 2023], which includes 96 scenarios across four room types. Each scenario contains 2–5 receptacles, with 4 example object placements per receptacle. The primary evaluation metric is *accuracy*, which measures the model's ability to predict the correct receptacle for a given object based on expected ground-truth placements.

Our Context-Oriented Benchmark. The TidyBot benchmark focuses on personalized preferences based on human demonstrations in relatively small scenes, where each object has only a single ground-truth placement. In contrast, our benchmark is designed to reflect object arrangement in larger, house-scale environments with greater spatial complexity and richer contextual cues.

We construct a new benchmark using over 20 scenes sampled from the HSSD 200 [Khanna et al. 2023] and APND [Ehsani et al. 2021] datasets. This collection spans more than 80 rooms, 200 functional areas, and 500 receptacle objects. Unlike existing benchmarks, which assign only one ground-truth placement per object, our benchmark acknowledges that in real household environments, objects often have multiple suitable placements, typically following a preference ranking that aligns with human intuition.

To support this, we introduce a novel annotation protocol and design evaluation metrics accordingly. We enlisted 10 trained annotators, each of whom remotely controlled a robot to navigate through every scene. With full access to scene-level information, annotators identified the most appropriate receptacles for each carriable object, considering both functional suitability and contextual relevance. This process ensures that the annotations reflect commonsense judgments rooted in practical use.

After collecting all annotations, we aggregated receptacle choices for each object using majority voting. This yielded a ranked list

Table 2. Comparison with Housekeep [Kant et al. 2022] and Tidybot [Wu et al. 2023] on Rearrangement and Misplacement detection task. we include several baseline strategies in the comparison to demonstrate improvement given by the different methods: *Random Selection*, *Pure LLM*, and *Pure LLM with Initial Scene Graph as Prompt* . The best results for each indicator are highlighted in bold.

Method	Rearrangement (Top k NDCG)								Misplacement detection			
	1	2	3	4	5	6	7	8	Accuracy	Recall	Precision	F1 score
Random	0.171	0.222	0.289	0.345	0.396	0.433	0.474	0.514	0.492	0.877	0.488	0.627
HouseKeep [Kant et al. 2022]	0.397	0.414	0.530	0.615	0.664	0.675	0.691	0.702	0.677	0.737	0.831	0.781
Tidybot [Wu et al. 2023]	0.573	0.629	0.669	0.706	0.734	0.758	0.779	0.798	0.796	0.890	0.800	0.842
Pure LLM	0.472	0.536	0.589	0.633	0.667	0.698	0.725	0.751	0.816	0.887	0.827	0.856
Pure LLM + Initial Scene Graph	0.570	0.612	0.661	0.692	0.715	0.740	0.767	0.788	0.802	0.879	0.828	0.853
Ours	**0.607**	**0.670**	**0.735**	**0.767**	**0.800**	**0.816**	**0.823**	**0.823**	**0.871**	**0.909**	**0.935**	**0.922**

Table 3. Ablation study on different context-induced types via evaluating the misplacement detection performance on our context-oriented dataset. The best results for each indicator are highlighted in bold.

Context-induced types			accuracy	recall	precision	F1
Textual	Visual	Global				
			0.817	0.893	0.860	0.876
✓			0.848	0.889	0.922	0.905
	✓		0.842	0.886	0.926	0.906
		✓	0.840	0.884	0.931	0.907
✓	✓		0.861	0.899	0.927	0.911
✓		✓	0.865	0.897	0.934	0.915
	✓	✓	0.865	0.897	0.935	0.915
✓	✓	✓	**0.871**	**0.909**	**0.935**	**0.922**

of valid placements per object, where the number of votes determines the ranking score. As a result, our benchmark supports a multi-ground-truth setting, enabling a more realistic and nuanced evaluation of object rearrangement systems, better aligned with the diversity and ambiguity of real-world environments.

Evaluation Metrics. We evaluate our method for the household rearrangement task from two key perspectives: *misplacement detection* and *rearrangement planning*. For misplacement detection, we generate over 10000 messy placements by randomly placing carriable objects across various scenes. The system is then tasked with identifying the misplaced items. Following standard evaluation protocols in the field [Padilla et al. 2020], we report *accuracy*, *recall*, *precision*, and *F1 score* to assess performance. To more comprehensively evaluate rearrangement planning, particularly under our benchmark setting where each object may have multiple ground-truth placements with human-defined rankings, we require methods to output a ranked list of recommended placements as well. For this, we adopt *Normalized Discounted Cumulative Gain* (NDCG@k) [Järvelin and Kekäläinen 2002], a widely used metric that evaluates how closely a predicted ranking aligns with the ground-truth ranking. NDCG effectively captures whether a method can prioritize placements consistent with human commonsense preferences. In our experiments, we consider the top 8 predicted placements and compute NDCG@1 to NDCG@8 scores for each carriable object.

For a fair comparison, all methods are evaluated under the same input and knowledge assumptions, using the same initial scene graph and benchmark-defined visual observations (i.e., a fixed RGB sequence in our benchmark and no visual input in the TidyBot benchmark). We use the open-source DeepSeek-V3-671B [Liu et al. 2024] as the language model and Qwen2.5-VL-72B [Bai et al. 2025] as the vision-language model in all LLM- and VLM-based experiments across both benchmarks for reproducibility. In our benchmark, we set both the number of generated queries per query type and the number of retrieved candidates passed to the decision planner to 10. To account for the inherent stochasticity of LLM outputs, we ran all methods three times and report the average results. We also add *token usage* and *failure rate* as metrics to measure the computational cost and hallucination issue of LLM in ablation study.

4.2 Quantitative Comparisons

Comparison on TidyBot Benchmark. We first compare our method with TidyBot [Wu et al. 2023] on its benchmark for the rearrangement planning task. TidyBot performs planning by imitating user-defined preferences, using two demonstration placements per receptacle during testing, where these demonstrations are summarized into a placing rule that is explicitly used to guide placement planning. In contrast, our method treats these demonstration placements as general contextual inputs for affordance analysis, rather than direct imitation targets. Despite this setting being less favorable to our approach, results in Table 1 show that our method outperforms TidyBot. This indicates that the Affordance-Enhanced Graph (AEG) effectively supports the LLM in extracting meaningful information, even when provided with sparse context.

Comparison on Our Context-Oriented Benchmark. We compare our method against both TidyBot [Wu et al. 2023] and Housekeep [Kant et al. 2022], a state-of-the-art learning-based rearrangement method that models object placement using the joint probability distribution of rooms, receptacles, and objects from a human-annotated dataset. Unlike our approach, Housekeep is trained by data with only a single ground-truth placement per object and does not incorporate functionality inferred from spatial relationships or object geometry. We also include random selection, pure LLM, and LLM prompted by initial scene graph as baselines for comparison. As shown in Table 2, our method significantly outperforms other baselines on the tasks of object rearrangement and misplacement detection. These results demonstrate that our proposed Affordance-Enhanced Graph (AEG) serves as an effective prompt structure, enabling the LLM to leverage rich contextual information beyond semantic labels alone, thereby yielding more accurate and context-aware predictions.

Table 4. Structural comparison between the initial scene graph (SG) and the Affordance Enhanced Graph (AEG), reporting the average number of relations per receptacle.

Average number of relations(edges) per receptacle in scene graph		
Type of Relations	Initial SG	AEG
Total relations	5.94	**9.09**
Spatial relations	5.94	5.94
Semantic relations	0.00	**5.88**
Updating existing relations	–	2.72
Introducing new relations	–	**3.22**
Cross-area relations	–	**0.92**

4.3 Ablation Studies

Comparison on Context-Induced Affordance. The core strength of our method lies in the Affordance-Enhanced Graph (AEG), constructed through context-induced affordance analysis. To evaluate the necessity of incorporating full contextual information, we compare our method against several alternatives with different levels of context integration: *no-context* (no context fusion or affordance analysis), *textual-context-enhanced scene graph*, *visual-context-enhanced scene graph*, and *global-context-enhanced scene graph*.

The results, presented in Table 3 and Figure 8 (top left), show that the vanilla scene graph fails to provide sufficient support for both misplacement detection and rearrangement planning, as it lacks the ability to reflect commonsense knowledge and contextual preferences. Among the individual context types, textual context is the most effective for both tasks. While global context contributes less on its own, it still improves performance when combined with other context types. Our full method, which integrates textual, visual, and global context into the affordance analysis, achieves the best overall results, demonstrating the necessity and effectiveness of comprehensive context fusion in scene understanding.

To better understand what global context contributes beyond downstream performance, we further analyze its structural impact on the scene graph. Table 4 summarizes a structural comparison between the initial scene graph and the Affordance Enhanced Graph (AEG). On average, global context introduces several semantic relations per receptacle comparable to the original local spatial relations. Notably, more than half of the introduced semantic relations establish new connections that do not exist in the initial scene graph, while the remaining ones refine existing spatial relations with richer functional semantics. In addition, a non-negligible portion of the semantic relations connect objects across different areas within the same room, indicating that global context enriches the graph with room-level functional relationships beyond local spatial proximity. These results help explain the stronger performance gains observed when incorporating global context into affordance analysis.

Comparison on Retrieval Augmented Generation. To validate the effectiveness of our rearrangement planning framework, we compare it against several baselines and ablated variants:

- **Ours without LLM filter:** Our method without the LLM-based filtering stage, relying solely on semantic similarity for candidate selection.

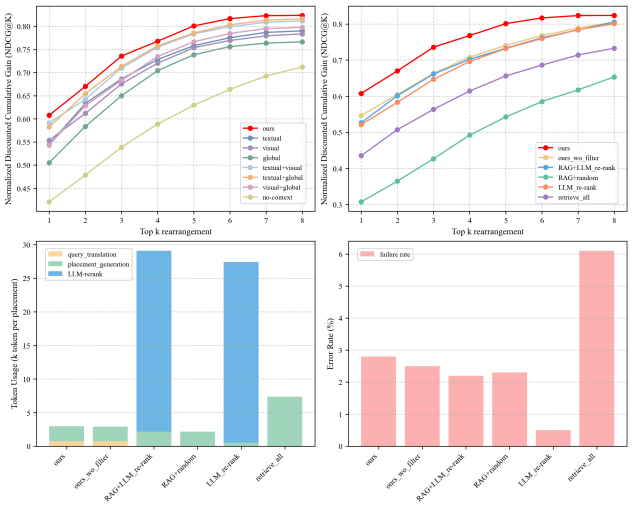

Fig. 8. Ablation study on context-induced affordance enhancement and Retrieval Augmented Generation via measuring the NDCG@k performance, token usage, and failure rate.

- **RAG + LLM re-rank:** Uses the LLM placement scorer to individually score and rank all receptacles, selecting the top-ranked candidates [Li et al. 2024].
- **RAG + random:** Randomly samples candidates.
- **LLM re-rank:** The LLM scores all receptacles and directly selects the top-scoring one as the final placement target, without a dedicated planning stage.
- **Retrieve all:** All receptacles in the scene are passed directly to the decision planner without any retrieval step.

The results, shown in Figure 8 (top right), demonstrate that even without the LLM-based filter, our method achieves performance comparable to the LLM re-ranking baseline. With the inclusion of the filter, our approach further improves and clearly outperforms all other baselines. The *Retrieve all* variant performs worse due to excessive prompt length in decision planning, which includes irrelevant information from unsuitable candidates and impairs the LLM's reasoning capability. The *RAG + random* variant performs the worst, as it frequently fails to retrieve relevant receptacle candidates, leading to unreliable planning outcomes.

We also evaluate each method's token usage and failure rate, with results presented in Figure 8 (bottom left and bottom right). Our method achieves the best overall performance while maintaining efficient token consumption. Although the *RAG + LLM re-rank* strategy delivers strong performance, it requires scoring all carriable–receptacle pairs individually and re-ranking them, resulting in high token usage per decision. Specifically, for each misplaced object, LLM re-ranking method must use the placement scorer to evaluate it against every receptacle in the scene. To complete rearrangement planning for an entire scene, this strategy incurs a total of $O(mn)$ LLM inference calls, where m is the number of misplaced objects and n is the number of receptacles. In contrast, our framework performs most of its computation offline during the database construction phase. At inference time, it only requires generating queries and a single decision-generation step per object, reducing

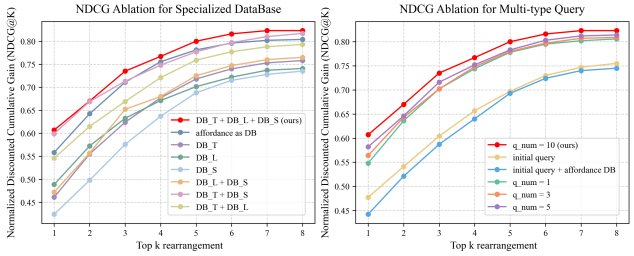

Fig. 9. Ablation study on Specialized Database and Multi-type Query via measuring the NDCG@k performance.

the LLM inference complexity to $O(m+n)$. This significantly lowers token consumption and enhances efficiency during deployment.

Failure rate is defined as the proportion of instances where the LLM fails to generate output in the required format. When this occurs, the system attempts a second invocation; if that also fails, it defaults to selecting the top-ranked candidate. Our method demonstrates an acceptable failure rate, indicating strong robustness and practical reliability for real-world deployment.

Comparison on Database Construction. To assess the necessity of incorporating all three databases, we compared our method against several variants, including a baseline that directly uses the context-induced affordance descriptions as a single unified database, as well as combinations of the Target Database DB_T, Location Database DB_L, and Scenario Database DB_S. As shown in Figure 8 (left), using only one of the three databases results in lower performance, while combining all three yields a clear performance gain and surpasses the affordance-only baseline. Among the individual databases, DB_T contributes the most, followed by DB_L, and then DB_S.

Comparison on Query Generation. We also assessed the impact of query design by comparing our translated query approach with two alternatives: (1) directly using the initial placement query to retrieve from the specialized databases, and (2) using the initial query to retrieve from the unified affordance-only database (i.e., naive RAG [Lewis et al. 2020]). As shown in Figure 8 (right), both alternatives perform notably worse than our approach, highlighting the importance of structured and context-aligned query translation. Additionally, we investigated the effect of varying the number of queries per type, finding that increasing *query_num* consistently improves performance—demonstrating the value of generating diverse query formulations for robust and comprehensive retrieval.

Comparison on Key Frame Selection. In this paper, we select a key frame for each object from an RGB sequence by calculating the total number of pixels occupied by the object and its neighbors in the image. To evaluate the impact of key frame selection for each node in the scene graph on affordance analysis, we conducted an ablation study focusing on different key frame selection strategies. We compared three methods that only consider the number of pixels occupied by the object itself in the image against the method discussed in the paper. These methods are:

- **Centering selection:** From all frames where the pixel count is not less than 100 pixels, select the frame where the object is closest to the center of the image.

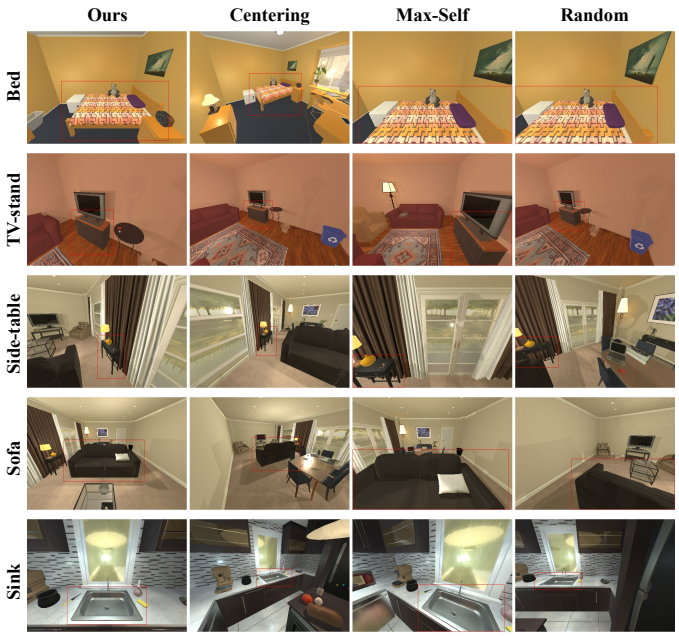

Fig. 10. Comparison of key frame selection for visual context extraction.

- **Max-self selection:** Select the frame where the object itself occupies the highest percentage of pixels in the image.
- **Random selection:** Choose a frame randomly from all frames where the pixel count is not less than 100 pixels.

The results, as shown in Table 5, indicate that our method performs the best. To further illustrate the influence of key frame selection on performance, we present several visual results in Figure 10. While the centering-selection method also aims to preserve as much context information of the object by positioning it at the center of the image, it often results in the object being farthest from the camera when centered. This leads to the object appearing very small in the selected key frame, and the overall density of information about the object and its surrounding context in the image is also reduced. Conversely, the max-self method ensures that the object occupies a significant portion of the image. However, this approach often results in the object appearing at the corner or on one side of the image, consequently missing much of the surrounding information.

Comparison on Area Construction. To assess the impact of different hierarchical construction strategies, we conducted an ablation study comparing our proposed area grouping method with three alternatives:

- **3D-position:** Clusters objects using the 3D position and orientation of their bounding box centers, rather than corner-to-corner distances.
- **Large-merge:** Uses the same corner-based distance matrix as our method but merges all clusters with fewer than four objects, potentially over-smoothing spatial partitions.

Table 5. Ablation result for key frame selection and area construction on Rearrangement and Misplacement detection task. The best results for each indicator are highlighted in bold.

Method		Rearrangement (Top k NDCG)								Misplacement detection			
		1	2	3	4	5	6	7	8	Accuracy	Recall	Precision	F1 score
Keyframe	random	0.556	0.620	0.693	0.740	0.777	0.788	0.796	0.799	0.857	0.892	0.922	0.906
	centering	0.586	**0.674**	0.726	0.767	0.797	0.810	0.818	0.820	0.865	0.896	0.935	0.915
	max-self	0.551	0.652	0.723	0.761	0.786	0.802	0.809	0.811	0.860	0.896	0.932	0.914
Area	3d-position	0.567	0.641	0.709	0.752	0.782	0.798	0.807	0.810	0.865	0.897	0.932	0.915
	large-merge	0.527	0.610	0.684	0.729	0.766	0.781	0.792	0.794	0.870	0.897	**0.936**	0.916
	no-merge	0.555	0.630	0.703	0.746	0.772	0.791	0.800	0.805	0.866	0.896	**0.936**	0.916
ours		**0.607**	0.670	**0.735**	**0.767**	**0.800**	**0.816**	**0.823**	**0.823**	**0.871**	**0.909**	0.935	**0.922**

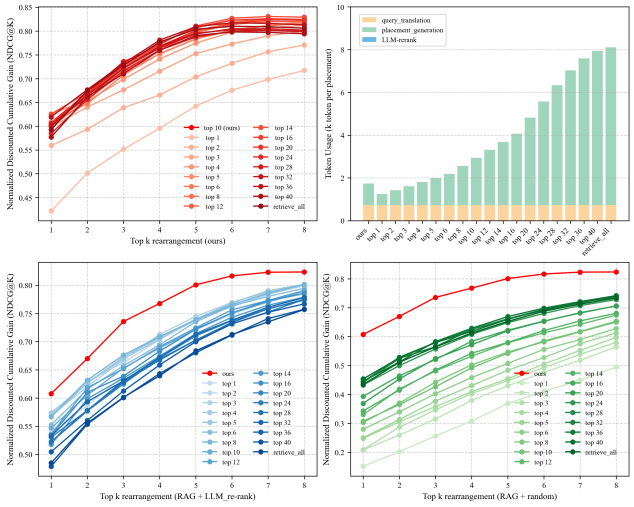

Fig. 11. Ablation study on Retrieve number of receptacles as a candidate via measuring the NDCG@k performance and token usage.

- **Random selection:** Applies the corner-based distance matrix without merging small clusters, resulting in many fragmented or single-object areas.

We applied affordance analysis using each grouping strategy and evaluated the resulting performance on misplacement detection and rearrangement planning. As shown in Table 5, our proposed method consistently outperforms the alternatives. Qualitatively, the *3D-position* method often misclusters large objects (e.g., beds, sofas) due to its reliance on center positions, ignoring spatial extent. The *Large-merge* method excessively aggregates distinct functional zones, collapsing sub-regions into overly large areas. Conversely, the *Random selection* approach produces over-fragmented layouts, with isolated single-object clusters that fail to capture functional relationships. Our method strikes a better balance by preserving semantic granularity while maintaining spatial coherence.

Comparison on Number of Candidates Retrieved. We also conducted an ablation study on the number of receptacle candidates retrieved during object rearrangement planning to evaluate its impact on placement quality. Specifically, we varied the retrieval size from 1 ∼ 40 and measured both NDCG performance and token usage. As shown in Figure 9 (top left and top right), performance initially improves significantly as the number of retrieved candidates increases, reaching a plateau around 10 and peaking between 10 ∼ 20. Beyond this range, performance becomes unstable and shows no consistent trend. In contrast, token usage per placement inference increases linearly with the number of retrieved candidates, as expected. Considering this trade-off between performance and efficiency, we select 10 as the final retrieval size for deployment.

We hypothesize that the observed performance fluctuations beyond this range may be influenced by the behavior of the LLM filter, which selectively eliminates unsuitable candidates. To better isolate the effect of retrieval size on placement quality, we disable the LLM filter and replace our retrieval mechanism with two alternative strategies: *LLM re-rank* and *random retrieval*. The results, shown in Figure 9 (bottom left and bottom right), reveal contrasting trends: the performance of LLM re-ranking declines as the retrieval size increases, while random retrieval shows a gradual improvement—though its best performance still lags behind LLM re-ranking. These findings indicate that good placement decision depends not on retrieving more candidates, but on retrieving highly relevant ones while filtering out irrelevant options. This targeted retrieval reduces noise in the prompt and enhances the decision quality.

We also conducted ablation studies, including the choice of LLM and VLM models, and prompt design for misplacement detection. Detailed results of these experiments are provided in the supplementary material.

4.4 Experiment on Scene Scalability

In our primary benchmark, all scenes contain a similar number of rooms and receptacles—typically around 40~50 receptacles per scene. As a result, the standard evaluation setting does not fully reflect our method's scalability in larger environments. To address this, we constructed larger composite scenes by combining existing scenes in our benchmark, simulating environments with a broader range of receptacle counts. Specifically, we generated ten composite scenes with the number of available receptacles ranging from 40 ∼ 400 and evaluated our method on each via measuring *NDCG@8* and token usage. For comparison, we included the baselines used in RAG ablation, enabling an assessment of each method's scalability.

As shown in Figure 12 (left), our method consistently outperforms all baselines across varying scene sizes. Notably, it does not

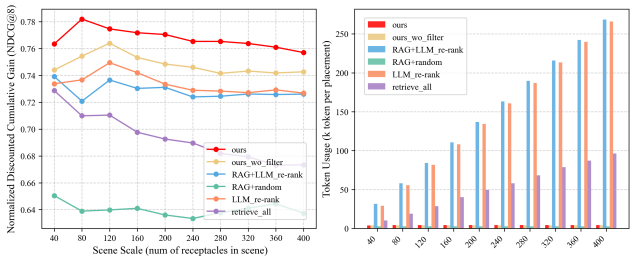

Fig. 12. scalability experiment for placement planning via measuring NDCG@8 & token usage.

exhibit significant performance degradation as the number of receptacles increases. In contrast, the *Retrieve All* method performs poorly in larger scenes, with a sharp performance drop due to excessive prompt length—caused by passing too many candidate receptacles into the decision planner—which overwhelms the LLM and impairs its reasoning ability. Figure 12 (right) further shows that our method maintains stable token usage per placement, even as the environment scales. By comparison, both methods containing *LLM re-rank* and *Retrieve All* incur rapidly increasing token costs in larger scenes. These results confirm that our method achieves strong scalability and generalization in complex, large-scale environments.

4.5 Qualitative Results

To better illustrate the pipeline of our method, we show some examples of context-induced affordance enhancement in Figure 14. Note how the agent derives more accurate functional descriptions of target objects than the original general semantic tags. Besides, we showcase several examples of carriable object placement planning based on AEG in Figure 15 with real-world scenarios and Figure 16 with synthetic environments. We can see that the agent can plan object placements that better fit the scene context based on AEG.

4.6 Real Robot Experiment

We deployed our algorithm and robotic system on a real-world robot to perform end-to-end household rearrangement. As illustrated in the Figure 13 , after placing the robot in a given scene, it first performs active exploration to complete a scan of the entire environment. The resulting RGB-D sequence is then used to construct the 3D scene graph and perform affordance enhancement. The LLM analyzes the affordance-enhanced graph to identify all misplaced objects and their appropriate placements. Using this information, the robot repeatedly navigates, grasps, and places the objects to complete the room rearrangement. The results demonstrate that our method is robust in real-world scenarios and is also token-efficient, enabling the robot to perform real-time scene enhancement and rearrangement planning. Additional results, including detailed token usage and an accompanying video, along with other applications of AEG, are available in the supplementary materials.

5 Conclusion

We introduced a method to automatically mine object functionalities aligned with user preferences from a scene by using multimodal foundation models prompted with scene graphs and RGB

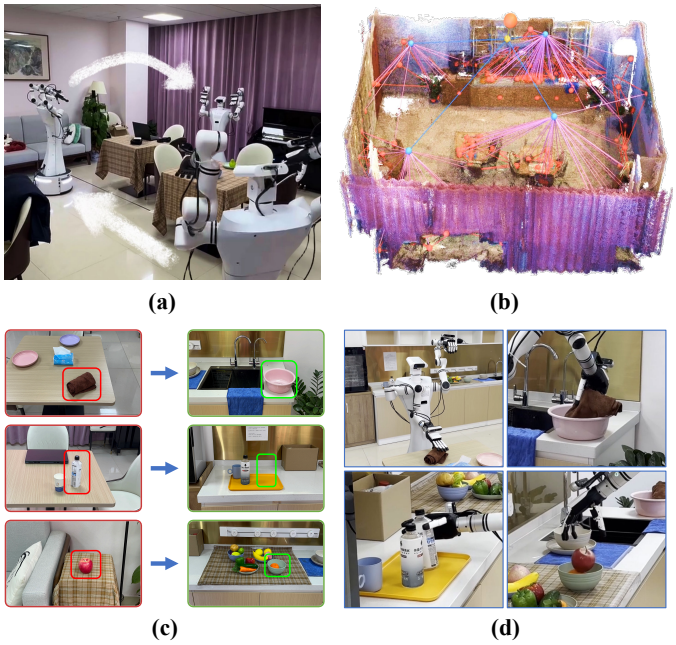

Fig. 13. Real-Robot Results for the End-to-End Household Rearrangement. (a) Active exploration in an unseen 3D scene. (b) Constructed Affordance-Enhanced Graph (AEG). (c) Detection of misplaced carriable objects and their placement targets. (d) Execution of the rearrangement task.

keyframes. The resulting affordance-enhanced graph effectively detects misplaced objects and determines their proper locations. Our method, along with the accompanying robotic system, has been validated in real-world environments, demonstrating that a robot can autonomously complete the entire rearrangement pipeline in previously unseen scenes—starting from exploration and SG construction from scratch, enhancing the SG into an AEG using LLM, and performing rearrangement task planning based on the AEG.

Our affordance enhancement, which integrates contextual information, is contingent upon the accuracy of the initial scene graph, thus, it cannot rectify errors inherent in its construction. Moving forward, we are keen on pursuing the joint optimization of the initial scene graph through end-to-end and/or active learning approaches.

At the same time, it is important to acknowledge the limitations of our current formulation. First, our method is built on the assumption that meaningful human preferences are implicitly reflected in the existing room layout and can be inferred from the observed scene context. As a result, in highly unstructured or random environments, or when the initial scene graph contains substantial errors, the inferred affordances and rearrangement decisions may become less reliable. Moreover, although the retrieval-augmented design significantly improves token efficiency and reduces hallucinations in LLM-based reasoning, it cannot eliminate such failures entirely, especially in rare or ambiguous scenarios. Addressing these limitations by jointly improving scene understanding, affordance reasoning, and robotic interaction capabilities remains an important direction for future research.

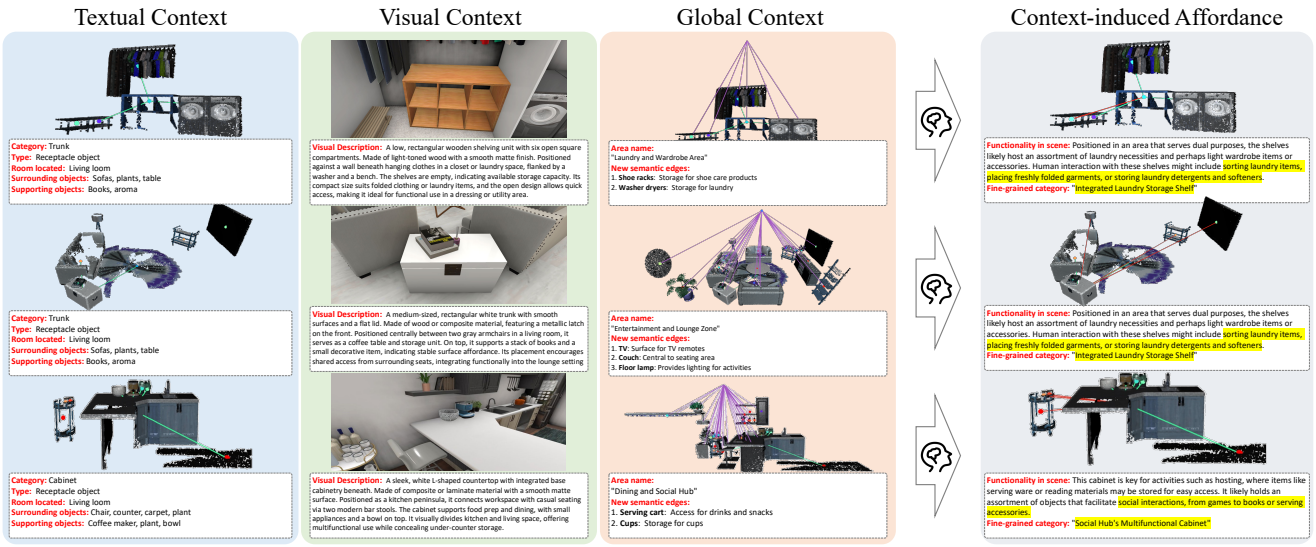

Fig. 14. Visual examples for Context Extraction, Context Fusion, and Affordance Analysis.

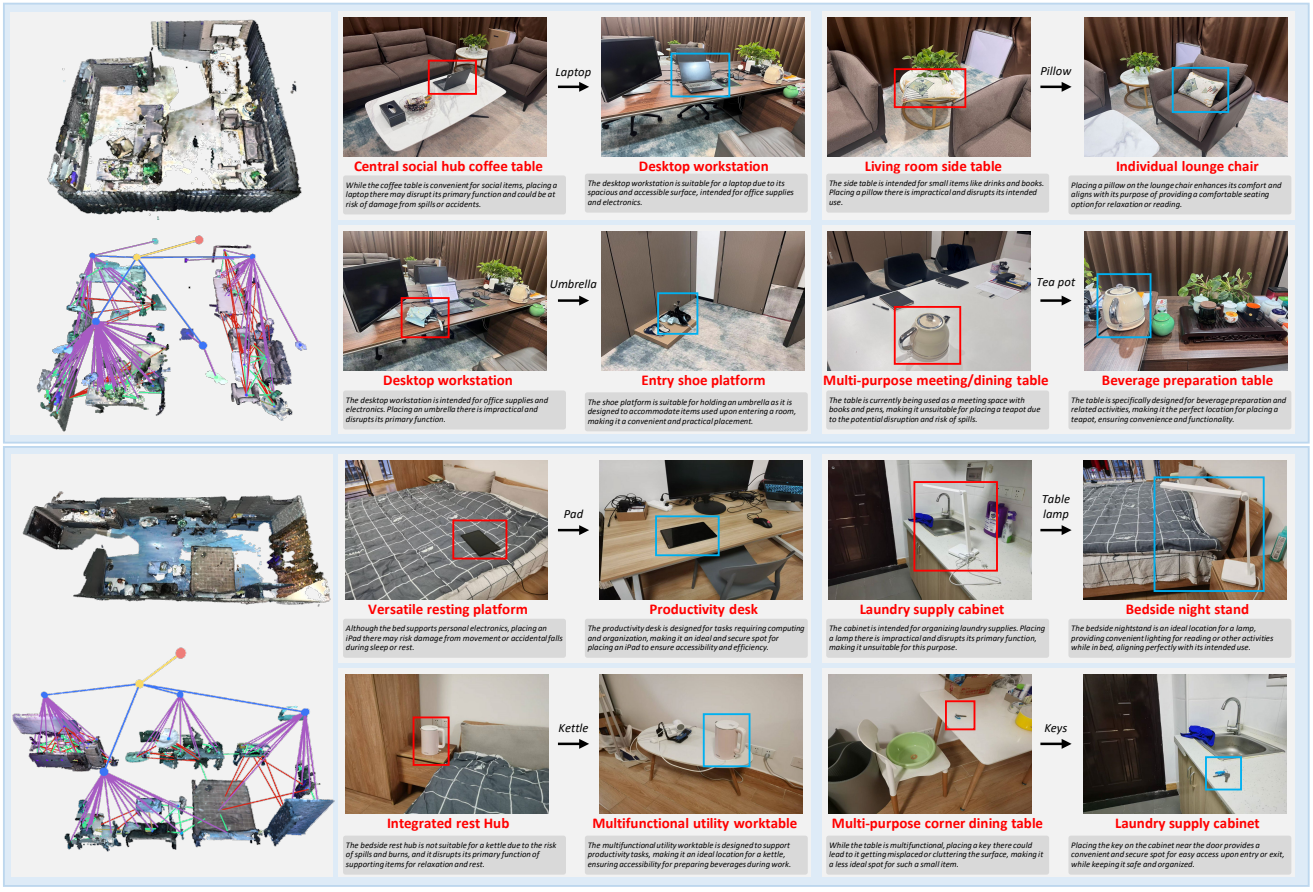

Fig. 15. Rearrangement planning examples in the real scenes, demonstrating eight rearrangement cases in two scenes. For example, the desk in the top scene is used as a workstation due to the placement of a monitor and laptop on top of it. Therefore, the umbrella on it is detected as a misplaced object and our method finds a more proper receptacle "Entry shoe Platform" for placement.

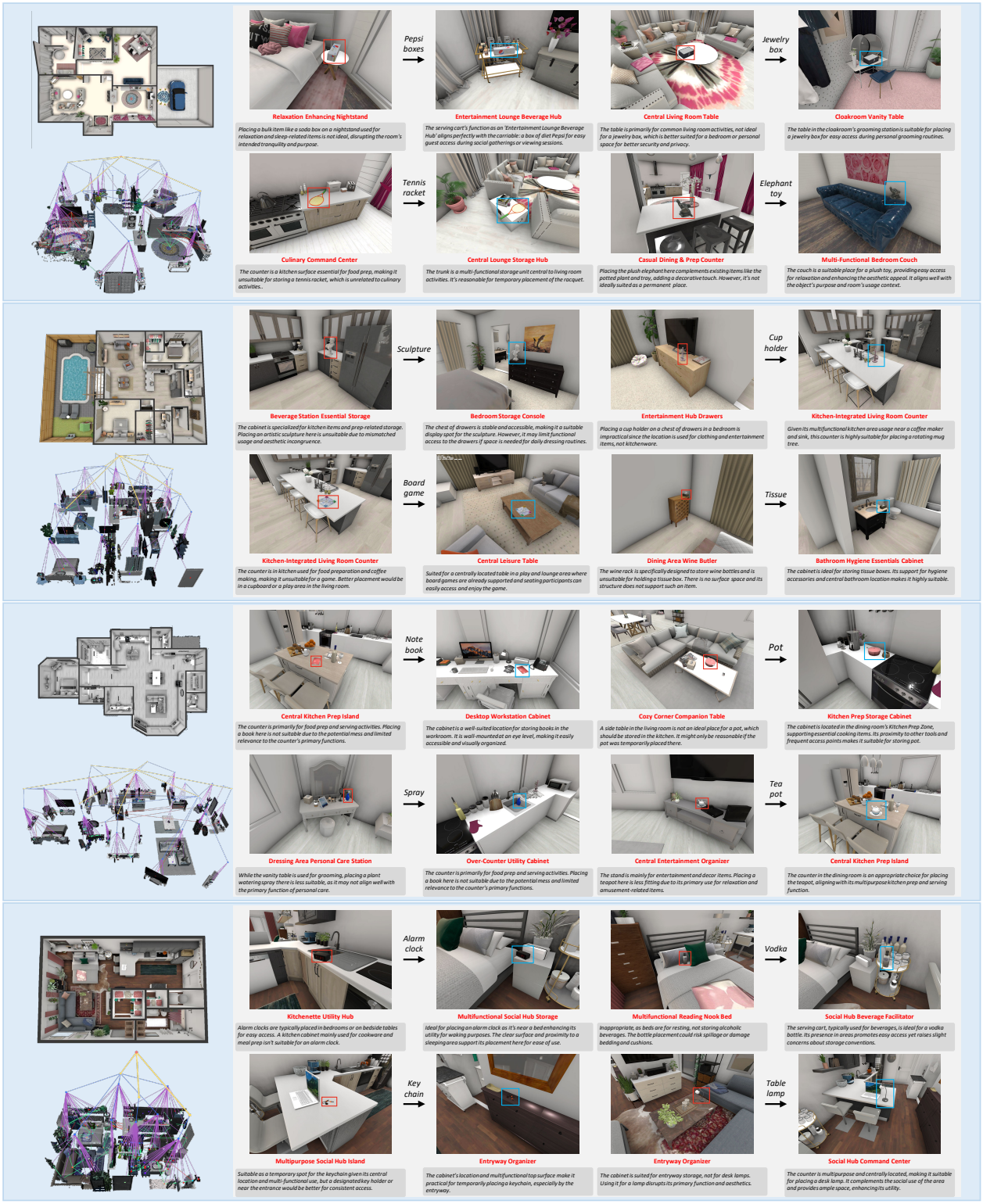

Fig. 16. Rearrangement planning examples in the synthetic scenes, demonstrating twelve rearrangement cases in three scenes. Here is an interesting example in the bottom scene, where our agent finds a better placement, the entryway organizer, for the key chain compared to its original location. The insight here is that our context analysis indicates that the desk is only suitable as a temporary spot for the keychain given its central location and multi-functional use. Therefore, our method searches the entry area of the scene and finds a more proper receptacle for placing the keychain.

Acknowledgments

This work was supported in part by the NSFC (62325211, 62132021, 62322207, 62522219, 62372457, 62572477), the Major Program of Xiangjiang Laboratory (23XJ01009).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

Panos Achlioptas, Ahmed Abdelreheem, Fei Xia, Mohamed Elhoseiny, and Leonidas Guibas. 2020. Referit3d: Neural listeners for fine-grained 3d object identification in real-world scenes. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer, 422–440.

Marcel R Ackermann, Johannes Blömer, Daniel Kuntze, and Christian Sohler. 2014. Analysis of agglomerative clustering. *Algorithmica* 69, 1 (2014), 184–215.

Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. 2022. Scanqa: 3d question answering for spatial scene understanding. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 19129–19139.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923* (2025).

Dhruv Batra, Angel X Chang, Sonia Chernova, Andrew J Davison, Jia Deng, Vladlen Koltun, Sergey Levine, Jitendra Malik, Igor Mordatch, Roozbeh Mottaghi, et al. 2020. Rearrangement: A challenge for embodied ai. *arXiv preprint arXiv:2011.01975* (2020).

Olvert A Berkhemer, Puck SS Fransen, Debbie Beumer, Lucie A Van Den Berg, Hester F Lingsma, Albert J Yoo, Wouter J Schonewille, Jan Albert Vos, Paul J Nederkoorn, Marieke JH Wermer, et al. 2015. A randomized trial of intraarterial treatment for acute ischemic stroke. *New England Journal of Medicine* 372, 1 (2015), 11–20.

Yihan Cao, Jiazhao Zhang, Zhinan Yu, Shuzhen Liu, Zheng Qin, Qin Zou, Bo Du, and Kai Xu. 2024. Cognav: Cognitive process modeling for object goal navigation with llms. *arXiv preprint arXiv:2412.10439* (2024).

Shengheng Deng, Xun Xu, Chaozheng Wu, Ke Chen, and Kui Jia. 2021. 3d affordancenet: A benchmark for visual object affordance understanding. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 1778–1787.

Yao Duan, Chenyang Zhu, Yuqing Lan, Renjiao Yi, Xinwang Liu, and Kai Xu. 2022. Disarm: displacement aware relation module for 3d detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16980–16989.

Kiana Ehsani, Winson Han, Alvaro Herrasti, Eli VanderBilt, Luca Weihs, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. 2021. Manipulathor: A framework for visual object manipulation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4497–4506.

Hao-Shu Fang, Chenxi Wang, Hongjie Fang, Minghao Gou, Jirong Liu, Hengxu Yan, Wenhai Liu, Yichen Xie, and Cewu Lu. 2023. Anygrasp: Robust and efficient grasp perception in spatial and temporal domains. *IEEE Transactions on Robotics* 39, 5 (2023), 3929–3945.

Kuan Fang, Te-Lin Wu, Daniel Yang, Silvio Savarese, and Joseph J Lim. 2018. Demo2vec: Reasoning object affordances from online videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2139–2147.

Matthew Fisher, Manolis Savva, Yangyan Li, Pat Hanrahan, and Matthias Nießner. 2015. Activity-centric scene synthesis for functional 3D scene modeling. *ACM Transactions on Graphics (TOG)* 34, 6 (2015), 1–13.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997* (2023).

Georgios Georgakis, Arsalan Mousavian, Alexander C Berg, and Jana Kosecka. 2017. Synthesizing training data for object detection in indoor scenes. *arXiv preprint arXiv:1702.07836* (2017).

James J Gibson. 1977. The theory of affordances. *Hilldale, USA* 1, 2 (1977), 67–82.

Helmut Grabner, Juergen Gall, and Luc Van Gool. 2011. What makes a chair a chair?. In *CVPR 2011*. IEEE, 1529–1536.

Abhinav Gupta, Scott Satkin, Alexei A Efros, and Martial Hebert. 2011. From 3d scene geometry to human workspace. In *CVPR 2011*. IEEE, 1961–1968.

Dongge Han, Trevor McInroe, Adam Jelley, Stefano V Albrecht, Peter Bell, and Amos Storkey. 2024. LLM-Personalize: Aligning LLM Planners with Human Preferences via Reinforced Self-Training for Housekeeping Robots. *arXiv preprint arXiv:2404.14285* (2024).

Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 2023. 3d-llm: Injecting the 3d world into large language models. *Advances in Neural Information Processing Systems* 36 (2023), 20482–20494.

Ruizhen Hu, Manolis Savva, and Oliver van Kaick. 2018. Functionality representations and applications for shape analysis. In *Computer Graphics Forum*, Vol. 37. Wiley Online Library, 603–624.

Dehao Huang, Chao Tang, and Hong Zhang. 2023. Efficient Object Rearrangement via Multi-view Fusion. *arXiv preprint arXiv:2309.08994* (2023).

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.

Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Keetha, Ayush Tewari, Joshua B. Tenenbaum, Celso Miguel de Melo, Madhava Krishna, Liam Paull, Florian Shkurti, and Antonio Torralba. 2023. ConceptFusion: Open-set Multimodal 3D Mapping. *Robotics: Science and Systems (RSS)* (2023).

Yash Kant, Arun Ramachandran, Sriram Yenamandra, Igor Gilitschenski, Dhruv Batra, Andrew Szot, and Harsh Agrawal. 2022. Housekeep: Tidying virtual households using commonsense reasoning. In *European Conference on Computer Vision*. Springer, 355–373.

Mukul Khanna, Yongsen Mao, Hanxiao Jiang, Sanjay Haresh, Brennan Shacklett, Dhruv Batra, Alexander Clegg, Eric Undersander, Angel X. Chang, and Manolis Savva. 2023. Habitat Synthetic Scenes Dataset (HSSD-200): An Analysis of 3D Scene Scale and Realism Tradeoffs for ObjectGoal Navigation. *arXiv preprint* (2023). arXiv:2306.11290 [cs.CV]

Hema S Koppula and Ashutosh Saxena. 2014. Physically grounded spatio-temporal object affordances. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part III 13*. Springer, 831–847.

Sumith Kulal, Tim Brooks, Alex Aiken, Jiajun Wu, Jimei Yang, Jingwan Lu, Alexei A Efros, and Krishna Kumar Singh. 2023. Putting people in their place: Affordance-aware human insertion into scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 17089–17099.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems* 33 (2020), 9459–9474.

Changyang Li, Haikun Huang, Jyh-Ming Lien, and Lap-Fai Yu. 2021. Synthesizing scene-aware virtual reality teleport graphs. *ACM Transactions on Graphics (TOG)* 40, 6 (2021), 1–15.

Gen Li, Deqing Sun, Laura Sevilla-Lara, and Varun Jampani. 2023. One-Shot Open Affordance Learning with Foundation Models. *arXiv preprint arXiv:2311.17776* (2023).

QI LI, Kaichun Mo, Yanchao Yang, Hang Zhao, and Leonidas Guibas. 2022. IFR-Explore: Learning Inter-object Functional Relationships in 3D Indoor Scenes. In *International Conference on Learning Representations*. https://openreview.net/forum?id=OT3mLgR8Wg8

Wenhao Li, Zhiyuan Yu, Qijin She, Zhinan Yu, Yuqing Lan, Chenyang Zhu, Ruizhen Hu, and Kai Xu. 2024. LLM-enhanced Scene Graph Learning for Household Rearrangement. In *SIGGRAPH Asia 2024 Conference Papers*. 1–11.

Xueting Li, Sifei Liu, Kihwan Kim, Xiaolong Wang, Ming-Hsuan Yang, and Jan Kautz. 2019. Putting humans in a scene: Learning affordance in 3d indoor environments. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 12368–12376.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437* (2024).

Weiyu Liu, Yilun Du, Tucker Hermans, Sonia Chernova, and Chris Paxton. 2023. StructDiffusion: Language-Guided Creation of Physically-Valid Structures using Unseen Objects. In *RSS 2023*.

Weiyu Liu, Chris Paxton, Tucker Hermans, and Dieter Fox. 2022. Structformer: Learning spatial structure for language-guided semantic rearrangement of novel objects. In *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 6322–6329.

Ziyuan Liu, Wei Liu, Yuzhe Qin, Fanbo Xiang, Minghao Gou, Songyan Xin, Maximo A Roa, Berk Calli, Hao Su, Yu Sun, et al. 2021. Ocrtoc: A cloud-based competition and benchmark for robotic grasping and manipulation. *IEEE Robotics and Automation Letters* 7, 1 (2021), 486–493.

Shiyang Lu, Haonan Chang, Eric Pu Jing, Abdeslam Boularias, and Kostas Bekris. 2023. Ovir-3d: Open-vocabulary 3d instance retrieval without training on 3d data. In *Conference on Robot Learning*. PMLR, 1610–1620.

Kaichun Mo, Yuzhe Qin, Fanbo Xiang, Hao Su, and Leonidas Guibas. 2022. O2O-Afford: Annotation-free large-scale object-object affordance learning. In *Conference on robot learning*. PMLR, 1666–1677.

Tushar Nagarajan, Christoph Feichtenhofer, and Kristen Grauman. 2019. Grounded human-object interaction hotspots from video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 8688–8697.

Tushar Nagarajan, Yanghao Li, Christoph Feichtenhofer, and Kristen Grauman. 2020. Ego-topo: Environment affordances from egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 163–172.

Toan Nguyen, Minh Nhat Vu, An Vuong, Dzung Nguyen, Thieu Vo, Ngan Le, and Anh Nguyen. 2023. Open-vocabulary affordance detection in 3d point clouds. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 5692–5698.

Zhe Ni, Xiao-Xin Deng, Cong Tai, Xin-Yue Zhu, Xiang Wu, Yong-Jin Liu, and Long Zeng. 2023. Grid: Scene-graph-based instruction-driven robotic task planning. *arXiv preprint arXiv:2309.07726* (2023).

Atsuyuki Okabe, Barry Boots, and Kokichi Sugihara. 1994. Nearest neighbourhood operations with generalized Voronoi diagrams: a review. *International Journal of Geographical Information Systems* 8, 1 (1994), 43–71.

Rafael Padilla, Sergio L Netto, and Eduardo AB Da Silva. 2020. A survey on performance metrics for object-detection algorithms. In *2020 international conference on systems, signals and image processing (IWSSIP)*. IEEE, 237–242.

Akshay Gadi Patil, Supriya Gadi Patil, Manyi Li, Matthew Fisher, Manolis Savva, and Hao Zhang. 2024. Advances in Data-Driven Analysis and Synthesis of 3D Indoor Scenes. In *Computer Graphics Forum*, Vol. 43. Wiley Online Library, e14927.

Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander William Clegg, Michal Hlavac, So Yeon Min, et al. 2023. Habitat 3.0: A co-habitat for humans, avatars and robots. *arXiv preprint arXiv:2310.13724* (2023).

Zackary Rackauckas. 2024. Rag-fusion: a new take on retrieval-augmented generation. *arXiv preprint arXiv:2402.03367* (2024).

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.

Abhinav Rajvanshi, Karan Sikka, Xiao Lin, Bhoram Lee, Han-Pang Chiu, and Alvaro Velasquez. 2023. Saynav: Grounding large language models for dynamic planning to navigation in new environments. *arXiv preprint arXiv:2309.04077* (2023).

Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian Reid, and Niko Suenderhauf. 2023. Sayplan: Grounding large language models using 3d scene graphs for scalable robot task planning. In *7th Annual Conference on Robot Learning*.

Tianhe Ren, Qing Jiang, Shilong Liu, Zhaoyang Zeng, Wenlong Liu, Han Gao, Hongjie Huang, Zhengyu Ma, Xiaoke Jiang, Yihao Chen, et al. 2024. Grounding dino 1.5: Advance the" edge" of open-set object detection. *arXiv preprint arXiv:2405.10300* (2024).

Gabriel Sarch, Zhaoyuan Fang, Adam W Harley, Paul Schydlo, Michael J Tarr, Saurabh Gupta, and Katerina Fragkiadaki. 2022. Tidee: Tidying up novel rooms using visuo-semantic commonsense priors. In *European conference on computer vision*. Springer, 480–496.

Manolis Savva, Angel X Chang, Pat Hanrahan, Matthew Fisher, and Matthias Nießner. 2014. SceneGrok: Inferring action maps in 3D environments. *ACM transactions on graphics (TOG)* 33, 6 (2014), 1–10.

Manolis Savva, Angel X Chang, Pat Hanrahan, Matthew Fisher, and Matthias Nießner. 2016. Pigraphs: learning interaction snapshots from observations. *ACM Transactions On Graphics (TOG)* 35, 4 (2016), 1–12.

Ketan Rajshekhar Shahapure and Charles Nicholas. 2020. Cluster quality analysis using silhouette score. In *2020 IEEE 7th international conference on data science and advanced analytics (DSAA)*. IEEE, 747–748.

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. 2012. Indoor segmentation and support inference from rgbd images. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V 12*. Springer, 746–760.

Chao Tang, Jingwen Yu, Weinan Chen, and Hong Zhang. 2021. Relationship oriented affordance learning through manipulation graph construction. *arXiv preprint arXiv:2110.14137* (2021).

Tuan Van Vo, Minh Nhat Vu, Baoru Huang, Toan Nguyen, Ngan Le, Thieu Vo, and Anh Nguyen. 2023. Open-vocabulary affordance detection using knowledge distillation and text-point correlation. *arXiv preprint arXiv:2309.10932* (2023).

Zan Wang, Yixin Chen, Baoxiong Jia, Puhao Li, Jinlu Zhang, Jingze Zhang, Tengyu Liu, Yixin Zhu, Wei Liang, and Siyuan Huang. 2024. Move as You Say, Interact as You Can: Language-guided Human Motion Generation with Scene Affordance. *arXiv preprint arXiv:2403.18036* (2024).

Zehan Wang, Haifeng Huang, Yang Zhao, Ziang Zhang, and Zhou Zhao. 2023. Chat-3d: Data-efficiently tuning large language model for universal dialogue of 3d scenes. *arXiv preprint arXiv:2308.08769* (2023).

Luca Weihs, Matt Deitke, Aniruddha Kembhavi, and Roozbeh Mottaghi. 2021. Visual room rearrangement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 5922–5931.

Jimmy Wu, Rika Antonova, Adam Kan, Marion Lepert, Andy Zeng, Shuran Song, Jeannette Bohg, Szymon Rusinkiewicz, and Thomas Funkhouser. 2023. TidyBot: Personalized Robot Assistance with Large Language Models. *Autonomous Robots* (2023).

Sriram Yenamandra, Arun Ramachandran, Karmesh Yadav, Austin Wang, Mukul Khanna, Theophile Gervet, Tsung-Yen Yang, Vidhi Jain, Alex William Clegg, John Turner, Zsolt Kira, Manolis Savva, Angel Chang, Devendra Singh Chaplot, Dhruv Batra, Roozbeh Mottaghi, Yonatan Bisk, and Chris Paxton. 2023. HomeRobot: Open Vocab Mobile Manipulation. https://aihabitat.org/static/challenge/home_robot_ovmm_2023/OVMM.pdf

Ceng Zhang, Xin Meng, Dongchen Qi, and Gregory S Chirikjian. 2024. RAIL: Robot Affordance Imagination with Large Language Models. *arXiv preprint arXiv:2403.19369* (2024).

Jiazhao Zhang, Liu Dai, Fanpeng Meng, Qingnan Fan, Xuelin Chen, Kai Xu, and He Wang. 2023. 3d-aware object goal navigation via simultaneous exploration and identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6672–6682.

Jiazhao Zhang, Chenyang Zhu, Lintao Zheng, and Kai Xu. 2020. Fusion-aware point convolution for online semantic 3d scene segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4534–4543.

Yixin Zhu, Chenfanfu Jiang, Yibiao Zhao, Demetri Terzopoulos, and Song-Chun Zhu. 2016. Inferring forces and learning human utilities from videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3823–3833.