

ASHiTA: Automatic Scene-grounded Hierarchical Task Analysis

Yun Chang* Leonor Fermoselle† Duy Ta‡ Bernadette Bucher‡ Luca Carlone* Jiuguang Wang†

Abstract

While recent work in scene reconstruction and understanding has made strides in grounding natural language to physical 3D environments, it is still challenging to ground abstract, high-level instructions to a 3D scene. High-level instructions might not explicitly invoke semantic elements in the scene, and even the process of breaking a high-level task into a set of more concrete subtasks—a process called hierarchical task analysis—is environment-dependent. In this work, we propose ASHiTA, the first framework that generates a task hierarchy grounded to a 3D scene graph by breaking down high-level tasks into grounded subtasks. ASHiTA alternates LLM-assisted hierarchical task analysis—to generate the task breakdown—with task-driven 3D scene graph construction to generate a suitable representation of the environment. Our experiments show that ASHiTA performs significantly better than LLM baselines in breaking down high-level tasks into environment-dependent subtasks and is additionally able to achieve grounding performance comparable to state-of-the-art methods.

1. Introduction

Modern machine vision applications, ranging from robotics to augmented reality, demand the development of vision systems that are able to parse and support the execution of high-level instructions, possibly provided by non-expert users. For example, consider a robot that is given a high-level task of preparing for dinner. To carry out this task, the robot is required to decompose the task into granular subtasks like fetching objects or visiting locations of interest (e.g., go to the kitchen, preheat the oven). Moreover, it must be able to ground these instructions into 3D objects and locations observed in the environment.

Towards these goals, prior work has mostly focused on developing more expressive map representations, which

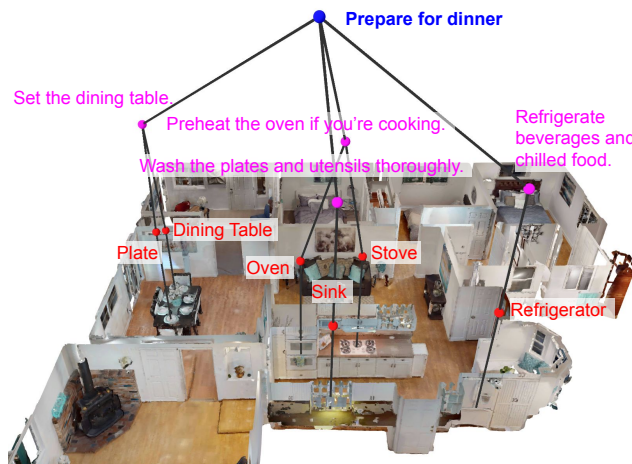


Figure 1. Given the high-level task of "Prepare for dinner", ASHiTA automatically generates a hierarchy of subtasks (and items) while grounding them to a 3D scene graph. For the scene graph in the figure, the blue node corresponds to the high-level task, magenta nodes correspond to the subtasks, and red nodes correspond to the items required by the subtasks.

combines both geometric and semantic information, to facilitate the grounding process. Related work investigates sparse object-based metric-semantic maps [36, 38], dense metric-semantic maps [31, 35], and 3D scene graphs [2, 15, 46]. Traditionally, these representations have relied on *closed-set semantic labels*, where the semantic concepts are restricted to the set of labels annotated in the training dataset. However, using a pre-defined set of labels limits the expressiveness of the map model (e.g., if our robot only has a "chair" class, it might not be able to distinguish a folding chair from a rocking chair), thus imposing constraints and creating ambiguity when grounding language commands.

To overcome this constraint, recent work has utilized new foundation models with *open-set semantics* for mapping [12, 17, 45]. However, as pointed out in [30], when using open-set semantics, it becomes unclear how to group objects (e.g., is the zipper of a bag an object, or only the bag itself?) Therefore, the work [30] proposes an approach that clusters open-set concepts in the map in a task-driven fashion, thus representing the objects and places in a scene graph at the granularity required by specified natural-language tasks. However, the resulting approach, Clío [30], still cannot support high-level tasks and is limited to supporting simple tasks that explicitly invoke relevant objects and locations (e.g., preheat the oven, go to the kitchen).

*LIDS, Massachusetts Institute of Technology, MA, USA. {yunchang, lcarlone}@mit.edu

†Robotics and AI Institute, MA, USA. {lfermoselle, dta, jw}@theaiinstitute.com

‡Department of Robotics, University of Michigan, MI, USA. bucherb@umich.edu

This work was done partially while Chang was an intern at the Boston Dynamics AI Institute and partially supported by the ARL DCIST Program.

Other approaches to solving complex robotics tasks requiring semantic reasoning try to sidestep this task-to-visual grounding problem. Side stepping this problem in tasks such as natural language navigation instruction following [13], semantic search problems [4, 49], and natural-language-directed pick and place [1, 11] leads to effective problem-specific solutions. More concretely, rather than rolling out the next action sequentially from what is currently seen by the robot’s camera [1, 13], we use explicitly stored observations from the entire scene for task decomposition, which enables the reasoning of tasks without clear termination conditions and over larger scenes.

Contribution. This paper presents the first framework to automatically generate a task hierarchy—which breaks down high-level tasks specified in natural language into subtasks and items—while grounding the task hierarchy into a 3D scene graph representation of the environment.

We start by considering the idealized case where the robot is not only given high-level tasks but also the grounded breakdown of the tasks into subtasks (what we call a *task hierarchy*). Our first contribution is to generalize the Information Bottleneck (IB) principle [42] used in [30] to generate a *hierarchy* of compressed representations: these are the layers of the 3D scene graph, which describe the scene at increasing levels of abstraction. We show that, under certain Markov assumptions between the random variables, we can derive an iterative algorithm for this hierarchical generalization of the IB (H-IB). Given a task hierarchy, the H-IB creates a scene graph where the layers are arranged according to the task hierarchy.

A fundamental issue is that in practice the robot is given the high-level task, but may not be given the task hierarchy (*i.e.* might not be provided with the details of the subtasks to execute). This is a common situation since the task hierarchy itself is environment-dependent. For instance, given the task “clean the office”, the exact breakdown and subtasks depend on the layout and items within the office in question. Therefore, the key observation is that not only should the map representation depend on the task, but the way we break down the task depends on the information in our map representation. Hence our second contribution is an iterative approach that alternates between a top-down hierarchical task analysis to generate a task hierarchy, and a bottom-up task-driven 3D scene graph construction to ground the task hierarchy. The result is ASHiTA, an approach for automatic scene-grounded hierarchical task analysis (Fig. 1).

We evaluate ASHiTA on both the idealized case with given task hierarchies and also the case where only high-level tasks are specified. Our experiments show that (i) given a task hierarchy, ASHiTA is able to generate a task-driven 3D scene graph that is more accurate in grounding task-relevant objects than state-of-the-art zero-shot visual-grounding models [52, 53] and (ii) given high-level natural

language tasks, ASHiTA is able to automatically generate a task hierarchy grounded to a 3D scene graph. To our knowledge, ASHiTA is the first system capable of accomplishing this, and our experiments demonstrate competitive advantages over naive LLM and scene graph baselines.

2. Related Work

Scene Graphs. Scene graphs have been used in computer graphics, vision, and robotics. 2D scene graphs such as [18] have been used for image retrieval, generation, and understanding. 3D scene graphs model 3D scenes using a graph where nodes are semantic concepts—grounded in 3D—and edges represent relations [2, 21, 35]. In contrast to voxel [28, 37] or neural feature field [20, 39] based representations, 3D scene graphs provide structured, relational information that directly supports high-level reasoning and task planning by explicitly capturing object relationships and attributes. SceneGraphFusion [46] estimates a scene graph of objects and relations in real-time. Hydra [14] is the first approach to build a hierarchical scene graph (including multiple layers) in real-time from sensor data. These algorithms and models focus on a closed set of semantic labels, which fundamentally limits their ability to represent novel and open-ended concepts encountered in complex, real-world environments.

Open-Set Semantic Understanding. A number of recent works have explored the use of vision-language foundation models for open-set semantic understanding. ConceptGraph [12] leverages vision-language models for 3D multi-view association, as well as SAM [22] and CLIP [33] to cluster a scene based on the objects’ semantic and geometric similarity. HOV-SG [45] leverages open-vocabulary vision foundation models to obtain segment-level maps in 3D and ultimately construct a hierarchy of concepts, though not in real time. CLIO [30] builds a task-driven 3D scene graph, where an information-theoretic framework is used to select the subset of objects and scene structures that are relevant to the task specified in natural language. While these approaches have made impressive strides towards semantically rich map representations, they still assume manually defined layers within the 3D scene graph, thus constraining the type of language instructions that can be grounded [30, 45].

Hierarchical Task Analysis. The breakdown of a complex long-horizon task into subtasks has been historically studied within the task and motion planning (TAMP) framework [10, 51]. Given a task in natural language, SayCan [3] uses LLMs to generate a set of feasible planning steps, re-scoring matched admissible actions with a learned value function, but limited by the size of the action space and the assumption of a 1 to 1 mapping between LLM output and action. LLM+P [27] uses an LLM as a PDDL writer in solving a planning problem described in natural language,

prompted with a domain PDDL and example problem-PDDL pairs as context. Similarly, ProgPrompt [41] leverages a programmatic LLM prompt structure to generate an entire executable plan program directly, functional across situated environments, robot capabilities, and tasks. AutoTAMP [6] uses Signal Temporal Logic (STL), derived from state observation and language instructions, as the intermediary representation and performs autoregressive re-prompting to correct syntactic and semantic errors. To enable geometrically and kinematically feasible plans, recent works have explored the use of intermediate goals instead of actions [48] and the generation of discrete and continuous language-parameterized constraints [24]. The ability of LLMs to perform task planning remains under debate, with both positive [40] and discouraging [43] results. Grounding the output of LLMs within the spatial and semantic structure of a scene graph [34] could potentially address the limitations of these approaches in their ability to integrate context-specific spatial information and actionable scene elements, which are essential for robust and contextually relevant task execution.

3. Problem Formulation

The core insight of ASHiTA is that when only given the visual observations of a 3D scene and high-level tasks, the proper breakdown of the tasks into subtasks depends on the available tools and objects in the scene representation, while at the same time, a scene representation capable of supporting the task execution depends on the task hierarchy.

Hierarchical Task Analysis. Hierarchical Task Analysis (HTA) is widely used in fields such as User Experience Design and Human-Computer Interaction [8, 32]. It is used to break down complex tasks into a task hierarchy composed of atomic subtasks to aid the understanding of how tasks are performed. In the context of this work, our goal is to perform HTA of a task that is grounded in the physical scene in order to understand how high-level tasks can be carried out with observed objects in a 3D environment.

Task-Driven Scene Graph. We construct a Task-Driven Scene Graph as a scene representation to ground the task hierarchy given by HTA. Each node in the scene graph corresponds to a task, a subtask, or an item involved in carrying out a subtask (Fig. 1). These nodes are organized hierarchically such that each task involves multiple subtasks, and each subtask involves interactions with certain items.

4. ASHiTA

This section describes ASHiTA, our approach for coupled HTA and task-driven 3D scene graph construction. The architecture is summarized in Fig. 2. ASHiTA consists of three main components: (i) construction of the primitives layer from visual input (Sec. 4.1), (ii) scene hierarchy up-

date (Sec. 4.2), where we construct and update the scene graph based on the task hierarchy for a given high-level task, and (iii) task update (Sec. 4.3), where the task hierarchy is updated based on the generated scene graph. Given a set of high-level tasks, ASHiTA starts with an initial task hierarchy and iteratively performs scene hierarchy update and task update to refine the task hierarchy and the scene graph in an alternating fashion.

4.1. Primitives Layer Construction

The primitives layer is the bottom layer of ASHiTA’s task-driven scene graph. It consists of a set of 3D bounding boxes with associated visual features. To construct the primitives layer from RGB-D inputs, a frontend performs primitive detection and association, and a backend jointly optimizes the estimated camera poses and primitive positions, ensuring spatial and temporal consistency of the primitives. The camera poses are queried from RTABMap [25].

The frontend relies on EfficientViT [29], a class-agnostic segmentation model, to detect class-agnostic 2D segments from the input RGB camera images and MobileCLIP [44], a visual-language encoder, to generate feature embeddings for each segment, thus forming a set of 2D primitives detected in the input frame. Note that the resolution of each primitive depends on the segmentation model, which means that a primitive can contain any object or object part. Each primitive is then projected to 3D and transformed to the world frame using the depth image and the estimated camera pose, forming the 3D primitives. To associate the latest 3D primitives with previously seen primitives, we first sample candidates from within a 1m radius, then formulate the problem of associating the latest primitives $I = \{i_0, \dots, i_N\}$ to the candidate primitives $J = \{j_0, \dots, j_M\}$ as an assignment problem using an $N \times M$ score matrix, where each entry contains the score $s(i, j)$ for matching primitive i with j . The score is defined as the weighted sum of visual similarity and semantic similarity between primitives i and j

$$s(i, j) = \omega_{text} \phi_{text}(i, j) + \omega_{visual} \sum_{bbox} k_{match}(i, j) \quad (1)$$

The visual similarity term is the sum of the match scores [26] of matched keypoints k_{match} [7] within the detection bounding box of primitives i and j , and the semantic similarity term is the cosine similarity ϕ_{text} of the feature embeddings of primitives i and j . We compute the optimal solution to this assignment problem with the Hungarian matching algorithm [23], by maximizing the total cumulative sum of scores for matching primitive i with primitive j . After association, we optimize the camera poses and primitive positions using GTSAM [9].

4.2. Scene Hierarchy Update

To construct the scene hierarchy from the primitives layer, we use and extend a well-known tool from information the-

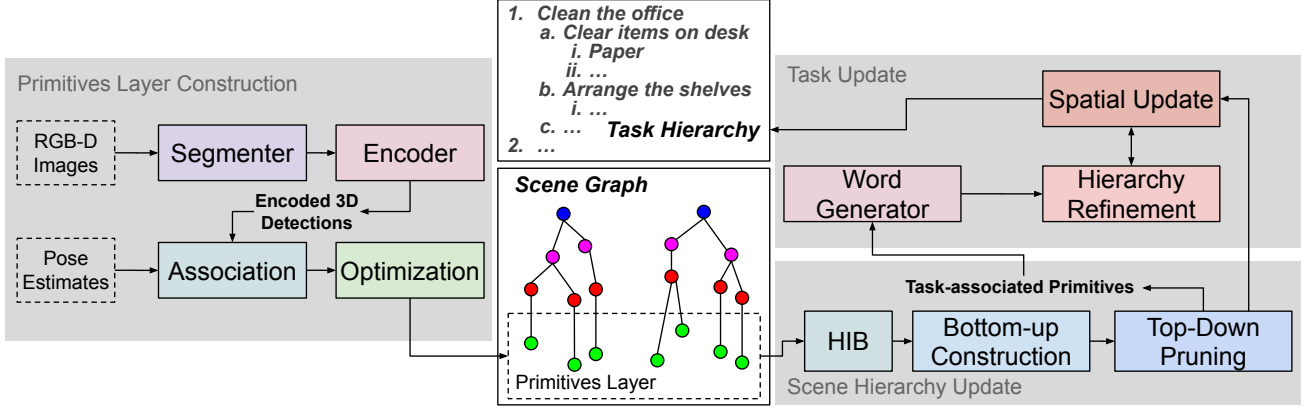


Figure 2. ASHiTA first segments and encodes primitives in 2D, and then associates and optimizes them in 3D together with the camera poses. ASHiTA then breaks down high-level tasks into a task hierarchy by alternating two steps: a *Scene Hierarchy Update* (Section 4.2) which creates a 3D scene graph from the primitives layer using the task hierarchy, and a *Task Update* (Section 4.3) which uses an LLM and the 3D scene graph to refine the task hierarchy.

ory, the *Information Bottleneck* [42], which compresses a representation in a task-driven fashion. In particular, we propose a hierarchical extension, dubbed the *Hierarchical Information Bottleneck* algorithm (H-IB), which compresses the primitives multiple times, according to the layers in the task hierarchy. We use the output of H-IB to first construct the scene graph in a bottom-up fashion, then perform a top-down pass to refine and prune the parts of the scene graph that have low task relevance.

Hierarchical Information Bottleneck. The Information Bottleneck algorithm (IB) [42] seeks to find a compressed representation \mathcal{S} of the input data \mathcal{S}_0 that retains as much relevant information about the task \mathcal{T} as possible while minimizing the amount of information about \mathcal{S}_0 that is not useful for \mathcal{T} . Intuitively, IB creates a "bottleneck" in the information flow between input \mathcal{S}_0 and task \mathcal{T} , and the compressed version of the input \mathcal{S} preserves the maximum amount of information about \mathcal{T} while discarding irrelevant details from \mathcal{S}_0 . Formally, this can be written as

$$\min_{\mathbb{P}(\mathcal{S}|\mathcal{S}_0)} I(\mathcal{S}_0; \mathcal{S}) - \beta I(\mathcal{T}; \mathcal{S}) \quad (2)$$

where $I(X; Y)$ is the mutual information between X and Y , and the parameter β controls the amount of compression.

H-IB is a generalization of IB to account for multi-resolution tasks $\mathcal{T}_1 \dots \mathcal{T}_n$ and multiple levels of compression $\mathcal{S}_1 \dots \mathcal{S}_n$. Conceptually, this can be visualized as passing input information through a series of bottlenecks of varying sizes—from large to small—each representing different levels of abstractions. Formally, we can write the new minimization functional as

$$\min_{\mathbb{P}(\mathcal{S}_k|\mathcal{S}_{k-1}), k=1 \dots n} \sum_{k=1}^n I(\mathcal{S}_{k-1}; \mathcal{S}_k) - \beta \sum_{k=1}^n I(\mathcal{T}_k; \mathcal{S}_k) \quad (3)$$

We solve for the minimum by taking the partial derivative of the Lagrangian of (3) with respect to $\mathbb{P}(\mathcal{S}_k|\mathcal{S}_{k-1})$.

Assuming the Markov chain conditions $\mathcal{T}_n \dots \leftarrow \mathcal{T}_1 \leftarrow \mathcal{S}_0 \leftarrow \dots \leftarrow \mathcal{S}_n$, we can write the Lagrangian in terms of known probabilities and $\mathbb{P}(\mathcal{S}_k|\mathcal{S}_{k-1})$. Setting the derivative to zero gives a set of multi-layer update steps

$$\begin{cases} p_\tau(\mathcal{S}_k|\mathcal{S}_{k-1}) = \frac{1}{\mathcal{Z}} p_\tau(\mathcal{S}_k) \exp(-\beta d) \\ p_{\tau+1}(\mathcal{S}_k) = \sum_{\mathcal{S}_{k-1}} p_\tau(\mathcal{S}_{k-1}) p_\tau(\mathcal{S}_k|\mathcal{S}_{k-1}) \\ p_{\tau+1}(\mathcal{T}_k|\mathcal{S}_k) = \sum_{\mathcal{S}_{k-1}} p_\tau(\mathcal{T}_k|\mathcal{S}_{k-1}) p_\tau(\mathcal{S}_{k-1}|\mathcal{S}_k) \end{cases} \quad (4)$$

where d is a weighted sum of the Kullback–Leibler Divergence D_{KL} across the multi-resolution tasks.

$$d = D_{KL}(p_\tau(\mathcal{T}_k|\mathcal{S}_k) || p_\tau(\mathcal{T}_k|\mathcal{S}_{k-1})) + \sum_{i=k+1}^n \sum_{\mathcal{S}_i} p_\tau(\mathcal{S}_i|\mathcal{S}_k) D_{KL}(p_\tau(\mathcal{T}_i|\mathcal{S}_i) || p_\tau(\mathcal{T}_i|\mathcal{S}_{k-1})) \quad (5)$$

Note that we recover the standard IB [42] when $n = 1$. The full derivation is given in the supplementary material.

As mutual information is always non-negative and $I(\mathcal{T}_k; \mathcal{S}_k)$ is bounded by the entropy $H(\mathcal{S}_k)$, the functional (3) is bounded from below, preventing it from decreasing indefinitely. Let us call \mathcal{C}_τ the objective function value at iteration τ in (3). Each iteration step is independent and reduces the objective with respect to one of the distributions $p(\mathcal{S}_k|\mathcal{S}_{k-1})$, $p(\mathcal{S}_k)$, and $p(\mathcal{T}_k|\mathcal{S}_k)$ while keeping the others fixed, therefore $\mathcal{C}_{\tau+1} \leq \mathcal{C}_\tau$. Since \mathcal{C}_τ is non-increasing and bounded below, the sequence converges. We remark that since the objective is not jointly convex in all distributions simultaneously, the algorithm may converge to different local minima depending on the initialization.

Bottom-Up Construction. Starting with an ungrounded initial task hierarchy, we generate a rough version of the scene graph using H-IB as sketched out in Fig. 3a. Starting with the primitives layer, we compute the normalized cosine similarities between the primitives (green boxes in Fig. 3a) and the text embedding of the items in the task hierarchy (red diamonds in Fig. 3a) and take all primitives that have

similarity of greater than 0.8 to form the input \mathcal{S}_0 to H-IB. The multi-level abstraction of the tasks corresponds to the task hierarchy such that \mathcal{T}_1 corresponds to the items in subtasks, \mathcal{T}_2 corresponds to the subtasks, and \mathcal{T}_3 corresponds to the high-level tasks. To account for the irrelevant parts of the scene, we augment our task hierarchy with a *null* task that consists of a single null action step with items: "item" and "thing" [19], shown as the gray diamonds in Fig. 3a.

H-IB takes as input the conditional probabilities $\mathbb{P}(\mathcal{T}_1|\mathcal{S}_0)$, $\mathbb{P}(\mathcal{T}_2|\mathcal{S}_0)$, and $\mathbb{P}(\mathcal{T}_3|\mathcal{S}_0)$. We compute these using the text embeddings of the items in the task hierarchy and the embeddings of the nodes in the primitives layer [30]. For \mathcal{T}_1 , we compute the cosine-similarities scores between each embedding f_t , $t \in \mathcal{T}_1$ and embedding f_s , $s \in \mathcal{S}_0$ and use the softmax function to obtain $\mathbb{P}(\mathcal{T}_1|\mathcal{S}_0)$. From this, we can compute the conditional probability of the subtasks $\mathbb{P}(\mathcal{T}_2|\mathcal{S}_0)$ for $t_2 \in \mathcal{T}_2$ and $s \in \mathcal{S}_0$.

$$p(t_2|s) = \sum_{t_1 \in \mathcal{T}_1} p(t_2|t_1)p(t_1|s) \quad (6)$$

and the conditional probability of the tasks $\mathbb{P}(\mathcal{T}_3|\mathcal{S}_0)$

$$p(t_3|s) = \sum_{t_2 \in \mathcal{T}_2} p(t_3|t_2)p(t_2|s) \quad (7)$$

The conditional probabilities are computed bottom-up and do not require the sentence embeddings for the subtask or task descriptions. This is in-line with the Markov chain assumption made when deriving for the H-IB iterations.

We solve H-IB with $\beta = 10$ and with a min iteration of 10 and a max iteration of 1000 or until convergence, defined as $C_\tau - C_{\tau+1} < 10^{-8}$. The output H-IB are the probabilities $\mathbb{P}(\mathcal{S}_1|\mathcal{S}_0)$, $\mathbb{P}(\mathcal{S}_2|\mathcal{S}_1)$, and $\mathbb{P}(\mathcal{S}_3|\mathcal{S}_2)$. Each conditional probability encodes a soft mapping from layer i to layer $i+1$ (i.e., the probability that an element in layer i is a child of an element in layer $i+1$). We take the $\arg \max$ of the conditional probabilities to obtain the inter-layer edges, corresponding to the highest probability mapping, between the primitives layer to the items, the items to the subtasks, and the subtasks to the tasks, respectively. We also map scene graph nodes to the highest probability task, subtask, or item in the task hierarchy by taking the $\arg \max$ of $\mathbb{P}(\mathcal{T}_k|\mathcal{S}_k)$. Intuitively, this step takes a set of primitives and funnel them through the task hierarchy to obtain a 3D scene graph that is aligned with the task hierarchy (Fig. 3a).

Top-Down Pruning. In some cases, H-IB produces multiple nodes associated to the same task entity (i.e., task, subtask, or item). In this step, we first merge any overlapping scene graph nodes associated to the same task entity and then discard irrelevant nodes top-down, as shown in Fig. 3b. To do this, we define *confidence* as the probability of a node s given the task entity t : $p(s|t)$. This can be easily computed from $p(t|s)$, obtained from H-IB, along with $p(s)$ and $p(t)$.

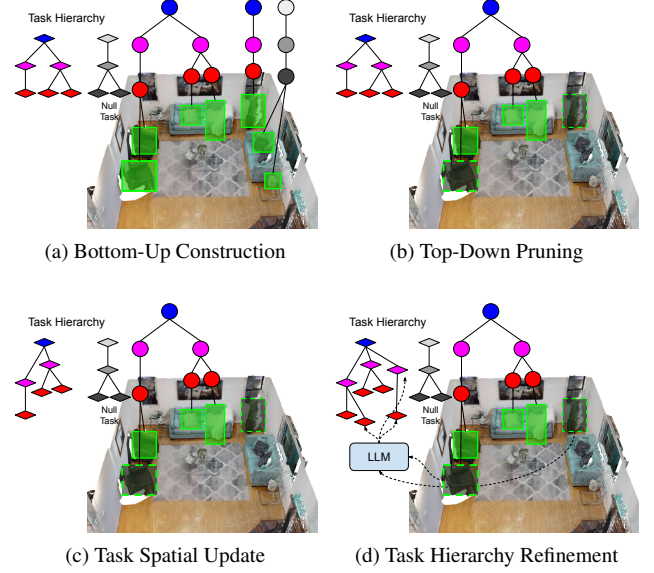


Figure 3. ASHiTA’s Scene Hierarchy and Task Update steps. The task hierarchy is on the left with diamond-shaped nodes representing the task entities. The scene graph is on the right, with circles marking the task-aligned scene graph nodes and green boxes marking the primitives layer. (a) Bottom-Up Construction: Starting from an initial task hierarchy, we perform H-IB and use the result to construct a 3D scene graph. (b) Top-Down Pruning: We perform pruning using the probabilities obtained from H-IB and also prune nodes related to the null tasks. (c) Spatial Update: Using the scene graph, we can update the spatial locations of the tasks, subtasks, and items. (d) Hierarchy Refinement: With the suggested items given by H-IB, we query the LLM to refine the task hierarchy.

Starting from the top of the task hierarchy, for each task, we compare all the associated nodes of the task and only keep the node with the highest confidence, pruning away all other nodes and their descendants (e.g. the deleted blue circles and its subtask and item nodes in Fig. 3b). We repeat this for the subtasks, and then also the items. Additionally, we prune all nodes and their descendants that are associated with the *null* task (e.g. deleted gray circles in Fig. 3b), as they are considered irrelevant to the given tasks. Lastly, we prune the primitives such that for each scene graph node on the items layer, we retain only the primitive with the highest confidence along with the primitives that intersect this highest confidence primitive. After pruning, we assign a bounding box and centroid to the item-associated scene graph node based on the union of the remaining primitives.

4.3. Task Update

The task hierarchy has to be both *spatially localized* (i.e., each subtask should be executed at a location in the map), and *environment-aware* (i.e., the task should reference elements in the environment). Hence, our task update consists of two steps: spatial update and hierarchy refinement.

Spatial Update. Initially, the task entities are not grounded, hence do not have any spatial information. Once a task entity in the task hierarchy is aligned with a scene

graph node, the task entity is *grounded*, and we update the spatial information of the task entity and its descendants accordingly, as sketched out in Fig. 3c. The position of a task entity is the centroid of the aligned scene graph node. The radius of a task entity is defined as the distance from the node to its nearest neighbor for the tasks and subtasks, or the Euclidean norm of its bounding box dimension for the items. This spatial approximation is accounted for in the input conditional probabilities for the next round for H-IB, as we will discuss below.

Hierarchy Refinement. The purpose of this step is to update the subtasks and items in the task hierarchy to account for primitives that are not fully aligned to the task hierarchy during Scene Hierarchy Update (Sec. 4.2), shown as the dark green boxes in Fig. 3d. This is done by tracking the subtasks and the primitives that are assigned to the subtasks during bottom-up construction (Fig. 3a), but are pruned during the top-down step (Fig. 3b). Recall that each primitive includes a visual embedding (Sec. 4.1). The Word Generator takes in these primitives and passes them through a word bank—an LLM-generated list of household items—and finds the items with the highest CLIP similarities. The output is a list of items present in the environment that can be used for each subtask. For each subtask, we query an LLM (Chat-GPT-4o-mini) to come up with a score between 0 and 1 for each item on the suggested list.

"Give me the probability values of each item in the list <suggestion> being required for the action: <subtask>."

We add the items with a score higher than a defined threshold r_s to the subtask, otherwise the suggested item is stored to query for new subtasks. Then, for each task, we collect the remaining suggested items not incorporated by its subtasks, and query an LLM for the likelihood scores in a similar manner, but with respect to the task. We collect the items with scores above a defined threshold r_t and query the LLM for additional subtasks as follows,

"Given the previous steps for <task>, add additional steps that involves only the following items: <suggested-items>. Make sure that the same item does not appear in more than one step."

The new subtasks are then added to the task hierarchy and incorporated in the next scene hierarchy update. This process is shown in Fig. 3d. In our experiments, we set $r_s = r_t = 0.8$ as the relevance threshold.

Spatially-Informed Conditional Probability. After task entities are grounded, they include spatial attributes, which need to be accounted for when computing the conditional probabilities for H-IB (Sec. 4.2). We define a spatial

Method	s-acc (%)	t-acc (%)
3D-VisTA [52]	25.3	10.3
PQ3D [53]	24.4	9.7
ASHiTA	28.71	12.13
ASHiTA + Txt Emb.	65.4	39.33
GPT w/ GT labels [50]	75.9	52.1

Table 1. Evaluation of sequential grounding using the SG3D HM3DSem dataset with ground truth 3D instance segmentation. Trials with knowledge of the ground truth object labels are highlighted.

conditional probability for primitive s_i and task entity t ,

$$p_s(s_i|t) = \eta \begin{cases} 1 & d < r \\ \exp(-(d-r)^2/r^2) & d \geq r; \end{cases} \quad (8)$$

where η is a normalization constant, d is the distance of a primitive to the item task entity, and r is the radius of the spatial attribute. We can then compute

$$p_s(t_k|s_i) = p_s(s_i|t_k)p(t_k)/p(s_i) \quad (9)$$

and define the new distribution as

$$p(t_k|s_i) = p_s(t_k|s_i)p_e(t_k|s_i)/\sum_k (p_s(t_k|s_i)p_e(t_k|s_i)) \quad (10)$$

where p_e is the embeddings-based conditional probability discussed in Sec. 4.2. These distributions are then used in the next round of the H-IB.

5. Experiments

In this section, we first evaluate the more straightforward case where the high-level tasks are already broken down into subtasks. We verify that ASHiTA is able to ground these given tasks and subtasks to objects in the scene (Sec. 5.1). Next, we evaluate the main contribution of this paper: the automatic generation of a task hierarchy grounded in a 3D scene graph given a high-level task (Sec. 5.2). To justify the design choices made in ASHiTA, we also perform an ablation study on the various components of ASHiTA (Sec. 5.3). Lastly, we qualitatively demonstrate the performance of ASHiTA in real-world scenes given various high-level tasks (Sec. 5.4).

5.1. Grounding Evaluation

We evaluate ASHiTA’s ability to ground subtasks to objects in the scene using the HM3DSem [47] test scenes with annotated task decomposition and grounding given by the SG3D dataset [50], which contains 890 high-level tasks distributed across 35 HM3DSem scenes.

To adapt ASHiTA for this evaluation, we use the given tasks and subtasks and generate an item for each subtask

Method	s-acc (%)	t-acc (%)
Hydra [14] + GPT	8.18	2.44
HOV-SG [45]	8.98	1.95
ASHiTA	21.7	8.78
Hydra (GT Seg) + GPT	14.2	6.34

Table 2. Evaluation of sequential grounding on 8 scenes from the SG3D HM3DSem dataset with incrementally built scene-graph representations. Trials with knowledge of the ground truth object labels are highlighted.

with Chat-GPT-4o-mini to form the initial task hierarchy. We only run half of the Hierarchy Refinement described in Sec. 4.3 to only update the items for each subtask and omit the step to add additional new subtasks.

We first evaluate the performance when given the ground truth 3D instance segmentation. We compare the performance of ASHiTA with image embeddings and ASHiTA with the text embeddings of the ground-truth labels against state-of-the-art zero-shot 3D visual grounding models (3D-VisTA [52] and PQ3D [53]) as benchmarked in [50]. We use two metrics in our evaluation [50]: subtask-accuracy (s-acc) and task-accuracy (t-acc) for given regions of the scene. The subtask-accuracy is the percentage of subtasks that are grounded to the correct object, and the task-accuracy is the percentage of tasks with subtasks that are all correctly grounded. The results are shown in Table 1. ASHiTA’s performance is significantly better than the baseline zero-shot models. The variants with ground truth information demonstrates performance without segmentation noise and with ideal visual-language alignment. In this limit, ASHiTA surpasses the best fine-tuned baseline in SG3D [50] and approaches the performance of GPT with ground-truth labels – falling short due to the lack of explicit relational information (e.g. “next to”, “on top of”) in ASHiTA.

To evaluate approaches better suited for embodied applications that incrementally build scene graphs, we also use 8 RGB-D sequences generated by [45] from 8 HM3DSem test scenes. This corresponds to 205 high-level tasks from the SG3D dataset [50]. Our baselines for this experiment are to use GPT to ground the task hierarchy on Hydra [15] and HOV-SG [45] generated scene graphs. For Hydra, we evaluate with both ground-truth 2D instance segmentation and with a closed-set segmentation model (OneFormer [16]), crop the resulting scene graph to the given evaluation region, then convert the resulting scene graph to a query-able format: {Room1: {Object 1: {label: <object-name>, positions: (x, y, z)}, Object2: ...}, Rom2: {}}. This is given to GPT-4o-mini alongside the tasks and the subtasks, and GPT identifies the node associated to each subtask. For HOV-SG, we also first crop the scene graph of the full scene to the given region, then we query the scene graph with the subtasks as described in [45] using GPT-4o-mini to retrieve the grounded object for each subtask. For evaluation, we require the centroid of the estimated object bounding

Method	s-rec (%)	s-prec (%)	t-acc (%)
Hydra + GPT	9.43	15.51	4.88
HOV-SG + GPT	4.55	4.87	1.95
ASHiTA	10.39	20.6	9.27
ASHiTA (GT Pos)	15.12	34.47	16.59
Hydra (GT Seg) [14] + GPT	17.06	18.98	14.63
ASHiTA (GT Pos + Txt Emb)	38.71	34.39	36.1

Table 3. Evaluation of Hierarchical Task Analysis. Gray highlight indicates directly using ground-truth object positions. Blue highlight marks trials having knowledge of ground-truth labels.

box to be contained by the ground-truth object bounding box to be considered a correct match. The results are shown in Table 2. We show that ASHiTA significantly outperforms the GPT-powered baselines, even when ground-truth segmentation is used in Hydra, cf. Hydra (GT Seg) in the table.

5.2. Hierarchical Task Analysis

We adapted the SG3D dataset [50] to evaluate hierarchical task analysis using the given subtasks as the reference task hierarchy for each high-level task. To evaluate a generated task hierarchy, we count an estimated subtask as *correct* if it grounds to the same set of objects as a reference subtask and *incorrect* otherwise. A reference subtask is *missed* if there is no estimated subtask that grounds to its set of objects. We report three metrics: subtask recall, step precision, and task accuracy. Given C correct subtasks, I incorrect subtasks, and M missed reference subtasks, the subtask recall (s-rec) is defined as $\frac{C}{C+M}$, and the subtask precision (s-prec) is defined as $\frac{C}{C+I}$. The task accuracy (t-acc) is the percentage of tasks with no incorrect subtasks.

We utilize the 8 RGB-D sequences [45] and the associated 205 high-level tasks for evaluation. We again use GPT-4o-mini and the scene graphs generated by Hydra [14] and HOV-SG [45] as baselines for evaluation. Instead of asking ChatGPT to identify the node associated to each subtask, we ask for the subtasks and nodes given the query-able scene graph and the high-level task. The results are shown in Table 3. ASHiTA outperforms the baselines by a large margin. The GPT-powered baselines tend to provide a large number of subtasks that include many contextually relevant but redundant and sometimes tangential subtasks, and this can be seen in particular with the lower subtask precision for Hydra with GPT. HOV-SG is not well suited to parsing high-level tasks, as it expects a formatted hierarchical query that is aligned with the scene graph structure.

5.3. System Ablations

We perform ablations on the components of ASHiTA, comparing the full ASHiTA pipeline against (i) replacing the H-IB with recursively running the original IB algorithm, (ii) removing the Top-Down pruning step (Sec. 4.2), (iii) removing the Spatial Update (Sec. 4.3), and (iv) remov-

Config	s-rec (%)	s-prec (%)	t-acc (%)
Recursive IB	1.51	24.53	1.46
w/o Top Down Pruning	9.22	18.93	5.37
w/o Spatial Update	8.7	22.22	6.34
w/o Hierarchy Refinement	7.71	23.13	6.83
Primitives + GPT	6.14	7.16	5.37
ASHiTA	10.39	20.6	9.27

Table 4. Ablation study on components of the ASHiTA pipeline.

ing the LLM-based Hierarchy Refinement (Sec. 4.3). We also include a comparison against using only GPT and the primitives layer: in this case, we first obtain words in the word bank based on the max cosine similarity score, and query GPT-4o-mini with the labeled primitives to come up with the subtasks and associated grounding. The results are shown in Table 4. While applying IB recursively compared to H-IB yields a higher precision, we obtain significantly lower recall due to the disconnect between layers of the task hierarchy in recursive IB variant.

5.4. Qualitative Results

Real-World Demonstrations. We demonstrate ASHiTA working in real-world environments. In Fig. 4, a Boston Dynamics Spot is teleoperated through a snack bar and an adjacent seminar room. Given high-level tasks of “prepare room for a seminar” and “prepare room for seminar reception”, ASHiTA was able to detect grounded items and subtasks associated with these high-level tasks. One limitation of ASHiTA exposed here is the ability to deal with multiple identical items. As seen under the chair arrangement subtask in Fig. 4, even though there are many identical chairs in the seminar room, ASHiTA retained only three chairs and assigned them unique labels. In Fig. 5, the Spot explores a hardware area in an office building. ASHiTA is given four high level tasks of varying abstraction, is able to perform reasonable hierarchical task analyses on the given tasks, and demonstrate the ability to incorporate additional subtasks to account for observed scene elements.

6. Limitations and Future Work

One major limitation of ASHiTA is the lack of relations in ASHiTA’s 3D scene graph, which means that ASHiTA is unable to accurately handle the incorporation of spatial specifications like “inside-of” or “on-top-of”. Additionally, the level of detail of the task hierarchy is limited by the descriptiveness of the word bank. Similar but distinct objects (e.g., different types of cups) will not be differentiated in the task hierarchy and the final scene graph if the word bank consists only of a general category type (e.g., cup). Lastly, there is no guarantee that the subtasks in the generated task hierarchy are sufficient to carry out the given high-level task. The hierarchical tree structure also constrains the

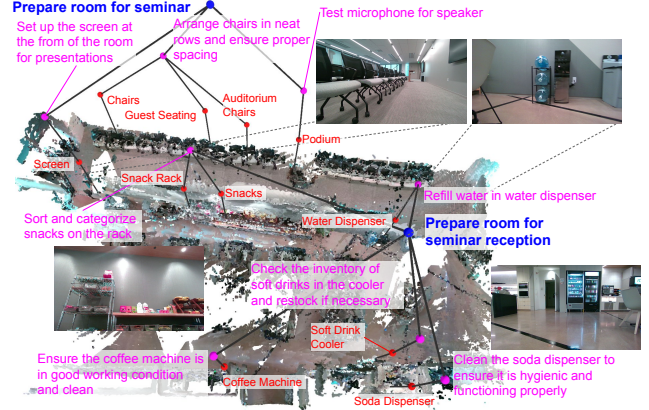


Figure 4. ASHiTA demonstrated in a real-world seminar room and snack bar on a robot, given two high-level tasks. Blue denotes the high-level tasks, magenta the decomposed subtasks, and red the items.

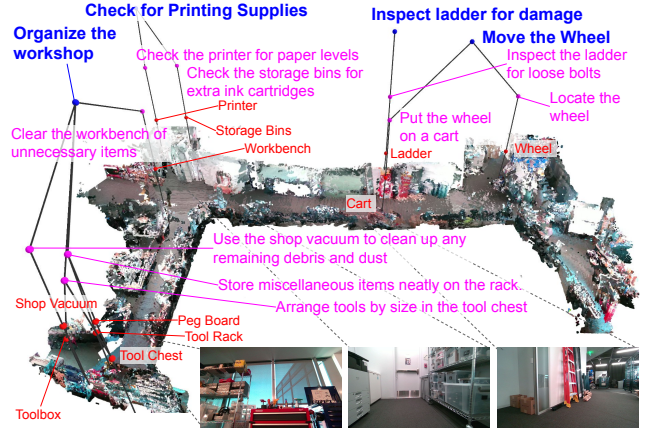


Figure 5. ASHiTA demonstrated in a real-world hardware workshop environment on a robot given 4 high-level tasks.

granularity of subtasks in an object-centric manner, preventing an object from being shared across multiple subtasks. A possible future direction is to incorporate classical planning approaches to verify and refine the generated subtasks [5].

7. Conclusion

We present ASHiTA, the first framework to generate a task hierarchy grounded to a 3D scene graph by breaking down high-level tasks into grounded subtasks. To achieve this, we introduce the Hierarchical Information Bottleneck (H-IB), a hierarchical generalization of the well known Information Bottleneck algorithm [42], and present an iterative approach that alternates between hierarchical task analysis and scene graph generation. We demonstrated that ASHiTA is able to decompose high-level tasks into grounded subtasks better than LLM and scene graph baselines by benchmarking against human-annotated scene-specific task breakdowns.

References

- [1] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. Do as i can and not as i say: Grounding language in robotic affordances. In *arXiv preprint arXiv:2204.01691*, 2022. 2
- [2] I. Armeni, Z. He, J. Gwak, A. Zamir, M. Fischer, J. Malik, and S. Savarese. 3D scene graph: A structure for unified semantics, 3D space, and camera. In *Intl. Conf. on Computer Vision (ICCV)*, pages 5664–5673, 2019. 1, 2
- [3] Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on Robot Learning (CoRL)*, pages 287–318. PMLR, 2023. 2
- [4] Junting Chen, G. Li, Suryansh Kumar, Bernard Ghanem, and Fisher Yu. How to not train your dragon: Training-free embodied object goal navigation with semantic frontiers. *Robotics: Science and Systems (RSS)*, 2023. 2
- [5] Y. Chen, J. Arkin, Y. Zhang, N.A. Roy, and C. Fan. Auto-TAMP: Autoregressive task and motion planning with LLMs as translators and checkers. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 6695–6702, 2023. 8
- [6] Yongchao Chen, Jacob Arkin, Yang Zhang, Nicholas A. Roy, and Chuchu Fan. Autotamp: Autoregressive task and motion planning with llms as translators and checkers. *arXiv preprintL 2306.06531*, 2023. 3
- [7] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-Supervised Interest Point Detection and Description. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 337–349, 2018. 3
- [8] D. Diaper and Neville A. Stanton. *The Handbook of Task Analysis for Human-Computer Interaction*. Lawrence Erlbaum, 2004. 3
- [9] F. Dellaert et al. Georgia Tech Smoothing And Mapping (GTSAM). <https://gtsam.org/>, 2019. 3
- [10] Caelan Reed Garrett, Rohan Chitnis, Rachel Holladay, Beomjoon Kim, Tom Silver, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Integrated task and motion planning. *Annual review of control, robotics, and autonomous systems*, 4(1):265–293, 2021. 2
- [11] Qiao Gu, Alihusein Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, Chuang Gan, Celso Miguel de Melo, Joshua B. Tenenbaum, Antonio Torralba, Florian Shkurti, and Liam Paull. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2024. 2
- [12] Qiao Gu, Alihusein Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, Chuang Gan, Celso Miguel de Melo, Joshua B. Tenenbaum, Antonio Torralba, Florian Shkurti, and Liam Paull. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2024. 1, 2
- [13] Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. A recurrent vision-and-language bert for navigation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2
- [14] N. Hughes, Y. Chang, and L. Carlone. Hydra: a real-time spatial perception engine for 3D scene graph construction and optimization. In *Robotics: Science and Systems (RSS)*, 2022. 2, 7, 1
- [15] N. Hughes, Y. Chang, S. Hu, R. Talak, R. Abdulhai, J. Strader, and L. Carlone. Foundations of spatial perception for robotics: Hierarchical representations and real-time systems. *Intl. J. of Robotics Research*, 2024. 1, 7, 2
- [16] Jitesh Jain, Jiachen Li, MangTik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. OneFormer: One Transformer to Rule Universal Image Segmentation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2989–2998, 2023. 7
- [17] 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. Conceptfusion: Open-set multimodal 3d mapping. In *Robotics: Science and Systems (RSS)*, 2023. 1
- [18] J. Johnson, R. Krishna, M. Stark, L. Li, D.A. Shamma, M.S. Bernstein, and L. Fei-Fei. Image retrieval using scene graphs. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3668–3678, 2015. 2
- [19] J. Kerr, C.M. Kim, K. Goldberg, A. Kanazawa, and M. Tancik. LERF: Language embedded radiance fields. In *Intl. Conf. on Computer Vision (ICCV)*, 2023. 5
- [20] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19729–19739, 2023. 2
- [21] U. Kim, J. Park, T. Song, and J. Kim. 3-D scene graph: A sparse and semantic representation of physical environments for intelligent agents. *IEEE Trans. Cybern.*, PP:1–13, 2019. 2
- [22] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Intl. Conf. on Computer Vision (ICCV)*, pages 4015–4026, 2023. 2
- [23] Harold W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics (NRL)*, 52, 1955. 3

- [24] Nishanth Kumar, Fabio Ramos, Dieter Fox, and Caelan Reed Garrett. Open-world task and motion planning via vision-language model inferred constraints. *arXiv preprint arXiv:2411.08253*, 2024. 3
- [25] Mathieu Labbé and François Michaud. Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. *J. of Field Robotics*, 36:416 – 446, 2018. 3
- [26] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed. *Intl. Conf. on Computer Vision (ICCV)*, pages 17581–17592, 2023. 3
- [27] Bo Liu, Yuqian Jiang, Xiaohan Zhang, Qiang Liu, Shiqi Zhang, Joydeep Biswas, and Peter Stone. Llm+p: Empowering large language models with optimal planning proficiency. *arXiv preprint: 2304.11477*, 2023. 2
- [28] Peiqi Liu, Zhanqiu Guo, Mohit Warke, Soumith Chintala, Chris Paxton, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Dynamem: Online dynamic spatio-semantic memory for open world mobile manipulation. *arXiv preprint arXiv:2411.04999*, 2024. 2
- [29] Xinyu Liu, Houwen Peng, Ningxin Zheng, Yuqing Yang, Han Hu, and Yixuan Yuan. Efficientvit: Memory efficient vision transformer with cascaded group attention. *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 14420–14430, 2023. 3
- [30] D. Maggio, Y. Chang, N. Hughes, M. Trang, D. Griffith, C. Dougherty, E. Cristofalo, L. Schmid, and L. Carlone. Clio: Real-time task-driven open-set 3D scene graphs. *IEEE Robotics and Automation Letters (RA-L)*, 2024. 1, 2, 5
- [31] J. McCormac, A. Handa, A. J. Davison, and S. Leutenegger. SemanticFusion: Dense 3D Semantic Mapping with Convolutional Neural Networks. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2017. 1
- [32] Donald A. Norman. *The Design of Everyday Things*. Basic Books, Inc., USA, 2002. 3
- [33] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Intl. Conf. on Machine Learning (ICML)*, pages 8748–8763. PMLR, 2021. 2
- [34] Aaron Ray, Christopher Bradley, Luca Carlone, and Nicholas Roy. Task and motion planning in hierarchical 3d scene graphs. *arXiv preprint arXiv:2403.08094*, 2024. 3
- [35] A. Rosinol, A. Violette, M. Abate, N. Hughes, Y. Chang, J. Shi, A. Gupta, and L. Carlone. Kimera: from SLAM to spatial perception with 3D dynamic scene graphs. *Intl. J. of Robotics Research*, 40(12–14):1510–1546, 2021. 1, 2
- [36] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. J. Kelly, and A. J. Davison. SLAM++: Simultaneous localisation and mapping at the level of objects. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013. 1
- [37] Nur Muhammad Mahi Shafiullah, Chris Paxton, Lerrel Pinto, Soumith Chintala, and Arthur Szlam. Clip-fields: Weakly supervised semantic fields for robotic memory. *arXiv preprint arXiv:2210.05663*, 2022. 2
- [38] M. Shan, Q. Feng, and N. Atanasov. Object residual constrained visual-inertial odometry. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 5104–5111, 2020. 1
- [39] Olaolu Shorinwa, Johnathan Tucker, Aliyah Smith, Aiden Swann, Timothy Chen, Roya Firoozi, Monroe David Kennedy, and Mac Schwager. Splat-mover: Multi-stage, open-vocabulary robotic manipulation via editable gaussian splatting. In *8th Annual Conference on Robot Learning*, 2024. 2
- [40] Tom Silver, Soham Dan, Kavitha Srinivas, Joshua B. Tenenbaum, Leslie Pack Kaelbling, and Michael Katz. Generalized planning in pdl domains with pretrained large language models. In *Nat. Conf. on Artificial Intelligence (AAAI)*, 2023. 3
- [41] Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. Progprompt: Generating situated robot task plans using large language models. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 11523–11530, 2023. 3
- [42] Naftali Tishby, Fernando Pereira, and William Bialek. The information bottleneck method. *Proc. of the Allerton Conference on Communication, Control and Computation*, 49, 2001. 2, 4, 8
- [43] Karthik Valmeekam, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. Planbench: An extensible benchmark for evaluating large language models on planning and reasoning about change. In *Conf. on Neural Information Processing Systems (NeurIPS)*, 2022. 3
- [44] Pavan Kumar Anasosalu Vasu, Hadi Pouransari, Fartash Faghri, Raviteja Vemulapalli, and Oncel Tuzel. Mobile-clip: Fast image-text models through multi-modal reinforced training. *arXiv preprint: 2311.17049*, abs/2311.17049, 2023. 3
- [45] Abdelrhman Werby, Chenguang Huang, Martin Büchner, Abhinav Valada, and Wolfram Burgard. Hierarchical open-vocabulary 3d scene graphs for language-grounded robot navigation. *Robotics: Science and Systems (RSS)*, 2024. 1, 2, 7
- [46] S. Wu, J. Wald, K. Tateno, N. Navab, and F. Tombari. Scene-GraphFusion: Incremental 3D scene graph prediction from RGB-D sequences. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 7515–7525, 2021. 1, 2
- [47] Karmesh Yadav, Santhosh Kumar Ramakrishnan, John Turner, Aaron Gokaslan, Oleksandr Maksymets, Rishabh Jain, Ram Ramrakhya, Angel X Chang, Alexander Clegg, Manolis Savva, Eric Undersander, Devendra Singh Chaplot, and Dhruv Batra. Habitat challenge 2022. <https://aihabitat.org/challenge/2022/>, 2022. 6, 1, 2
- [48] Zhutian Yang, Caelan Garrett, Dieter Fox, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. Guiding long-horizon task and motion planning with vision language models. *arXiv preprint arXiv:2410.02193*, 2024. 3
- [49] N. Yokoyama, S. Ha, D. Batra, J. Wang, and B. Bucher. VLFM: Vision-language frontier maps for zero-shot seman-

- tic navigation. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2024. [2](#)
- [50] Zhuofan Zhang, Ziyu Zhu, Pengxiang Li, Tengyu Liu, Xiaojian Ma, Yixin Chen, Baoxiong Jia, Siyuan Huang, and Qing Li. Task-oriented sequential grounding in 3d scenes. *arXiv preprint: 2408.04034*, abs/2408.04034, 2024. [6](#), [7](#), [2](#), [3](#)
- [51] Zhigen Zhao, Shuo Cheng, Yan Ding, Ziyi Zhou, Shiqi Zhang, Danfei Xu, and Ye Zhao. A survey of optimization-based task and motion planning: from classical to learning approaches. *IEEE/ASME Transactions on Mechatronics*, 2024. [2](#)
- [52] Ziyu Zhu, Xiaojian Ma, Yixin Chen, Zhidong Deng, Siyuan Huang, and Qing Li. 3d-vista: Pre-trained transformer for 3d vision and text alignment. *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2899–2909, 2023. [2](#), [6](#), [7](#)
- [53] Ziyu Zhu, Zhuofan Zhang, Xiaojian Ma, Xuesong Niu, Yixin Chen, Baoxiong Jia, Zhidong Deng, Siyuan Huang, and Qing Li. Unifying 3d vision-language understanding via promptable queries. *arXiv preprint: 2405.11442*, abs/2405.11442, 2024. [2](#), [6](#), [7](#)