UniGoal: Towards Universal Zero-shot Goal-oriented Navigation

Hang Yin[1]*, Xiuwei Xu[1*†], Linqing Zhao[1], Ziwei Wang[2], Jie Zhou[1], Jiwen Lu[1‡]
[1]Tsinghua University
[2]Nanyang Technological University
{yinh23, xxw21, zhaolinqing}@mails.tsinghua.edu.cn;
ziwei.wang@ntu.edu.sg; {jzhou, lujiwen}@tsinghua.edu.cn

Abstract

In this paper, we propose a general framework for universal zero-shot goal-oriented navigation. Existing zero-shot methods build inference framework upon large language models (LLM) for specific tasks, which differs a lot in overall pipeline and fails to generalize across different types of goal. Towards the aim of universal zero-shot navigation, we propose a uniform graph representation to unify different goals, including object category, instance image and text description. We also convert the observation of agent into an online maintained scene graph. With this consistent scene and goal representation, we preserve most structural information compared with pure text and are able to leverage LLM for explicit graph-based reasoning. Specifically, we conduct graph matching between the scene graph and goal graph at each time instant and propose different strategies to generate long-term goal of exploration according to different matching states. The agent first iteratively searches subgraph of goal when zero-matched. With partial matching, the agent then utilizes coordinate projection and anchor pair alignment to infer the goal location. Finally scene graph correction and goal verification are applied for perfect matching. We also present a blacklist mechanism to enable robust switch between stages. Extensive experiments on several benchmarks show that our UniGoal achieves state-of-the-art zero-shot performance on three studied navigation tasks with a single model, even outperforming task-specific zero-shot methods and supervised universal methods. Project Page.

1. Introduction

Goal-oriented navigation is a fundamental problem in various robotic tasks, which requires the agent to navigate to a specified goal in an unknown environment. Depending on the goal type, there are many popular sub-tasks of goal-oriented navigation, among which we focus on three representative types: object category, instance image and text description. These sub-tasks are also known as Object-goal Navigation (ON) [7, 44, 49], Instance-image-goal Navigation (IIN) [15, 16] and Text-goal Navigation (TN) [37]. With the development of deep learning, reinforcement learning (RL) and vision/language foundation models, we have witnessed great achievement of performance on each individual sub-task. However, in actual application scenarios, the high flexibility of human instructions requires high versatility of agent. Therefore, a universal method that can handle all sub-tasks in a single model is of great desire.

Towards the aim of universal goal-oriented navigation, a natural solution is to learn a uniform representation of different kinds of goals. GOAT [6] trains a universal global policy

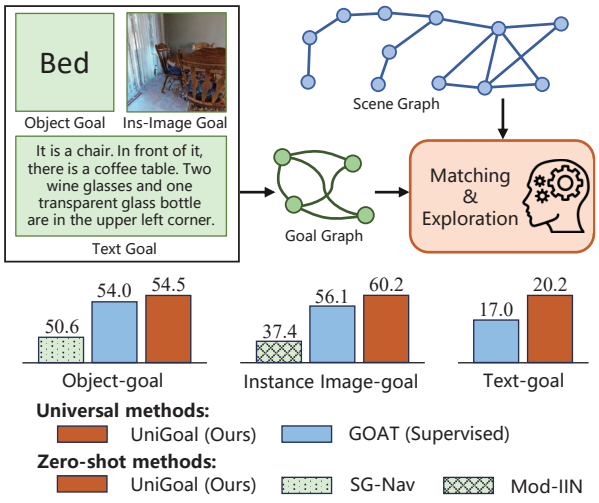

Figure 1. State-of-the-art zero-shot goal-oriented navigation methods are typically specialized for each goal type. Although recent work presents universal goal-oriented navigation method, it requires to train policy networks on large-scale data and lacks zero-shot generalization ability. We propose **UniGoal**, which enables zero-shot inference on three studied navigation tasks with a unified framework and achieves leading performance on multiple benchmarks.

* Equal contribution. † Project lead. ‡ Corresponding author.

on the three goal-oriented sub-tasks, which learns a shared goal embedding with RL. To reduce the requirement of training resources, PSL [37] utilizes CLIP [28] embedding to uniformly represent category, image and text description. But it still requires time-consuming RL training for the policy networks. Moreover, these training-based methods tend to overfit on the simulation environment and thus show weak generalization ability when applied to real world. To solve above limitations, zero-shot navigation methods [44, 49] appear to be an ideal choice, where the agent does not require any training or finetuning when deployed to a certain task. The mainstream solution of zero-shot methods is to leverage large language models (LLM) [1, 39] for general reasoning and decision making. However, although utilizing LLM makes zero-shot navigation feasible, current methods are designed for specific sub-task, which cannot transfer to wider range of goal types. The recent InstructNav [25] proposes a general framework to solve several language-related navigation tasks with chain-of-thought, but it is still unable to handle vision-related navigation like IIN. Therefore, a uniform inference framework for universal zero-shot goal-oriented navigation is highly demanded.

In this paper, we propose UniGoal to solve the above problems. Different from previous works which represent scene and goal in text format and design task-specific workflow for LLM, we propose a uniform graph representation for both 3D scene and goal and formulate a general LLM-based scene exploration framework. With our graph-based representation, the 3D scene, object category, instance image and text description can be uniformly represented with minimal structural information loss compared to text description. The consistent graph format between scene and goal also enables accurate explicit reasoning including similarity computation, graph matching and graph alignment. Specifically, we construct an online 3D scene graph along with the moving of agent. At each time instant, we first conduct graph matching between the scene graph and goal graph. Then we propose a multi-stage scene exploration policy, which adopts different strategies to generate long-term goal of exploration according to different matching states. With exploration of unknown regions, the matching score will increase and the policy will progress between three stages: iterative subgraph searching for zero matching, coordinate projection and anchor pair alignment for partial matching, and scene graph correction and goal verification for perfect matching. To enable robust switch between stages, we also present a blacklist mechanism to freeze unmatched parts of graphs and encourage exploration to new regions. Experimental results on several benchmarks of MatterPort3D [4], HM3D [30] and RoboTHOR [10] show that UniGoal achieves superior performance on all three tasks with a single model, even outperforming zero-shot methods that designed for specific task and universal methods that requires training or finetuning.

2. Related Work

Zero-shot Navigation. Conventional supervised navigation methods [7, 11, 18, 31, 41] requires large-scale training in simulation environments, which limits the generalization ability. According to the goal type, zero-shot navigation can be mainly divided into ON [3, 46], IIN [16] and TN [25]. Based on open-vocabulary CLIP [28], CoW [12] constructs zero-shot ON baseline using frontier-based exploration (FBE). ESC [49], OpenFMNav [8] and VLFM [45] further extract common sense about correlations between objects using LLM for goal location reasoning. For zero-shot IIN, Mod-IIN [16] simply utilizes FBE for exploration and key point matching for goal identification. For TN, currently there are only supervised methods [6, 37] and zero-shot ones are still missing. The recent InstructNav [25] proposes a universal zero-shot framework for language-related navigation tasks. It can be applied to ON, demand-driven navigation (DDN) and vision-language-navigation (VLN) [9, 20, 26, 47, 48]. But it is still unable to handle visual goal as in IIN. Different from these approaches, our UniGoal proposes a unified graph representation for goal and scene, which can elegantly handle both language and vision-related goal-oriented navigation tasks.

Graph-based Scene Exploration. In order to better understand and explore the scene, there are a variety of scene representation methods. Among them, graph-based representation [2, 13, 14, 33, 40] is one of the most popular and promising representations based on explicit graph structure, which shows great potential to be combined with LLM and VLM for high-level reasoning. SayPlan [32] leverages LLM to perform task planning by searching on a 3D scene graph. OVSG [5] constructs an open-vocabulary scene graph for context-aware descriptions and performs graph matching to ground the queried entity. There are also many works using graph representation for navigation tasks [24, 29, 42, 44]. Among them the most related work to ours is SG-Nav [44], which constructs an online hierarchical scene graph and proposes chain-of-thought prompting for LLM to reason the goal location based on graph sturcture. However, SG-Nav is specialized for ON, thus cannot fully exploit the rich information contained in other types of goal. While our UniGoal further represents the goal in a graph and perform graph matching between scene and goal to guide a multi-stage scene exploration policy, which make full use of the correlation between scene and goal for LLM reasoning.

3. Approach

In this section, we first provide the definition of universal zero-shot goal-oriented navigation and introduce the framework of UniGoal. Then we construct graphs for scene and goal and conduct graph matching for goal identification. Next we design a multi-stage scene exploration policy by

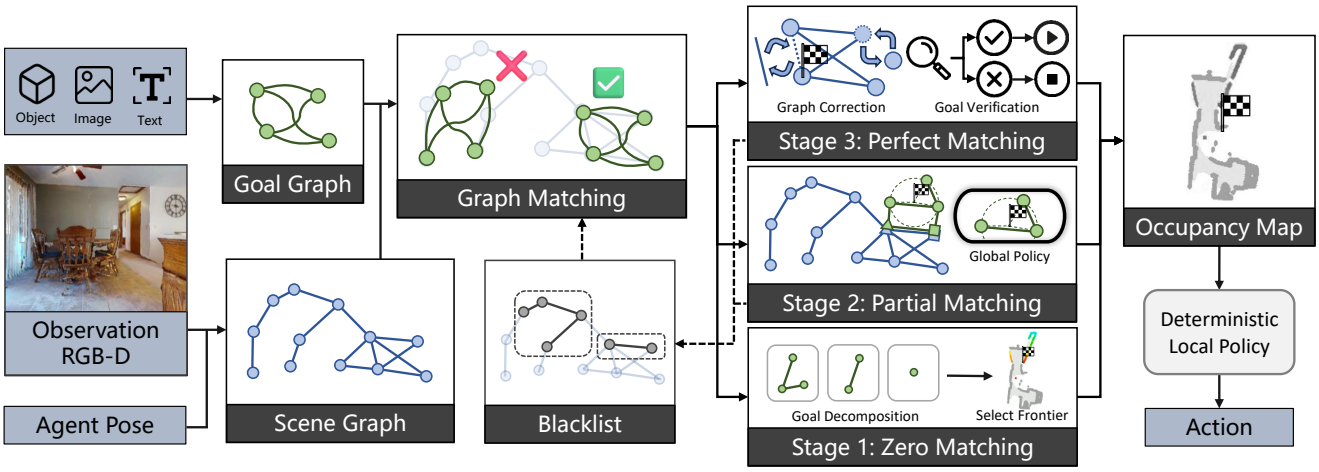

Figure 2. Framework of UniGoal. We convert different types of goals into a uniform graph representation and maintain an online scene graph. At each step, we perform graph matching between the scene graph and goal graph, where the matching score will be utilized to guide a multi-stage scene exploration policy. For different degree of matching, our exploration policy leverages LLM to exploit the graphs with different aims: first expand observed area, then infer goal location based on the overlap of graphs, and finally verify the goal. We also propose a blacklist that records unsuccessful matching to avoid repeated exploration.

prompting LLM with graphs. Finally we propose a blacklist mechanism to robustly avoid repeated exploration.

3.1. Goal-oriented Navigation

In goal-oriented navigation, a mobile agent is tasked with navigating to a specified goal g in an unknown environment, where g can be an object category (i.e. Object-goal Navigation [7], ON), an image containing object that can be found in the scene (i.e. Instance-Image-goal Navigation [15], IIN), or a description about a certain object (i.e. Text-goal Navigation [37], TN). For IIN and TN, there is a central object o as well as other relevant objects in g. While for ON, we have $o = g$. The agent receives posed RGB-D video stream and is required to execute an action $a \in \mathcal{A}$ at each time it receiving a new RGB-D observation. $\mathcal{A}$ is the set of actions, which consists of `move_forward`, `turn_left`, `turn_right` and `stop`. The task is successfully done if the agent stops within r meters of o in less than T steps. More details about the three sub-tasks can be found in supplementary material.

Task Specification. We aim to study the problem of universal zero-shot goal-oriented navigation, which has two characteristics: (1) Universal. We should design a general method, which requires no modification when switching between the three sub-tasks. (2) Zero-shot. All three kinds of goal can be specified by free-form language or image. Our navigation method does not require any training or finetuning, which is of great generalization ability.

Overview. Universal zero-shot goal-oriented navigation requires the agent to complete different sub-tasks with a single framework in training-free manner. Since this task requires extremely strong generalization ability, we utilize large language model (LLM) [1, 39] for zero-shot decision making by exploiting its rich knowledge and strong reasoning ability. To make LLM aware of the visual observations as well as unifying different kinds of goals, we propose to represent the scene and goal in graphs, i.e., *scene graph* and *goal graph*. In this way, different goals are represented uniformly and the representations of scene and goal are consistent. Based on this representation, we prompt LLM with scene graph and goal graph for scene understanding, graph matching and decision making for exploration. The overall pipeline is illustrated in Figure 2.

3.2. Graph Construction and Matching

In this subsection, we first describe how to construct a uniform graph representation for scene and goal. Then we propose a graph matching method to determine whether a goal or its relevant objects are observed, which further guides the selection of scene exploration policies.

Graph Construction. We define graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ as a set of nodes $\mathcal{V}$ connected with edges $\mathcal{E}$. Each node represents an object. Each edge represents the relationship between objects, which only exists between spatially or semantically related object pairs. The content of nodes and edges is described in text format. Since the agent is initialized in unknown environment and continuously explores the scene, we follow SG-Nav [44] to construct the scene graph $\mathcal{G}_t$ incrementally by expanding it every time the agent receives a new RGB-D observation. For goal graph $\mathcal{G}_g$, we adopt different methods to process three kinds of goals g into a graph, which we detail in supplementary material.

Graph Matching. With scene graph $\mathcal{G}_t$ and goal graph $\mathcal{G}_g$, we can match these two graphs to determine whether the goal is observed. If no elements in $\mathcal{G}_g$ are observed, the agent needs to infer relationship between objects from $\mathcal{G}_t$ to plan a path that is most likely to find the goal. If $\mathcal{G}_g$ is partially observed in $\mathcal{G}_t$, the agent can use the overlapped part of $\mathcal{G}_g$ and $\mathcal{G}_t$ to reason out where the rest of $\mathcal{G}_g$ is. If $\mathcal{G}_g$ is perfectly observed in $\mathcal{G}_t$, the agent can move to the goal and make further verification. Therefore, a goal scoring method is crucial for the follow-up scene exploration.

We propose to apply graph matching to achieve this. Given $\mathcal{G}_t$ and $\mathcal{G}_g$, we design three matching metrics, i.e., node matching, edge matching and topology matching, to score how well the goal is observed. Formally, for nodes and edges, we extract their embeddings and then compute pair-wise similarity with bipartite matching to determine matched pairs of nodes and edges:

$$\mathcal{M}_N = \mathcal{B}(\text{thr}(\text{Embed}(\mathcal{V}_t) \cdot \text{Embed}(\mathcal{V}_g)^T)) \quad (1)$$

$$\mathcal{M}_E = \mathcal{B}(\text{thr}(\text{Embed}(\mathcal{E}_t) \cdot \text{Embed}(\mathcal{E}_g)^T)) \quad (2)$$

where $\text{Embed}(\cdot) \in \mathbb{R}^{K \times C}$, K is the number of nodes or edges and C is the channel dimension, which is detailed in supplementary material. $\text{thr}(\cdot)$ is an element-wise threshold function applied on the similarity matrix which sets values smaller than τ to -1 to disable matching of the corresponding pairs. $\mathcal{B}(\cdot)$ is bipartite matching, which outputs a list of all matched node or edge pairs, namely $\mathcal{M}_N$ or $\mathcal{M}_E$. We also average the similarity matrix of nodes and edges to acquire the similarity scores S_N and S_E.

Based on $\mathcal{M}_N$ and $\mathcal{M}_E$, we further compute topological similarity between $\mathcal{G}_t$ and $\mathcal{G}_g$, which is defined as the graph editing similarity between them:

$$S_T = 1 - \mathcal{D}(\mathcal{S}(\mathcal{F}(\mathcal{G}_t, \mathcal{M}_N, \mathcal{M}_E)), \mathcal{S}(\mathcal{G}_g)) \quad (3)$$

where $\mathcal{F}(\mathcal{G}_t, \mathcal{M}_N, \mathcal{M}_E)$ means the minimal subgraph of $\mathcal{G}_t$ with nodes in $\mathcal{M}_N$ and edges in $\mathcal{M}_E$. $\mathcal{S}(\cdot)$ means the topological structure of a graph regardless of the content of nodes and edges. $\mathcal{D}(\cdot)$ is the normalized editing distance [34] between two graphs. The final matching score is defined as $S = (S_N + S_E + S_T)/3$.

3.3. Multi-stage Scene Exploration

As described above, different matching scores will lead to different scene exploration policies. From zero matching to perfect matching, we design three stages to progressively explore the scene and generate long-term exploration goal. This long-term goal will be processed by a deterministic local policy [36] to obtain actions. Below we detail our exploration policy stage by stage.

Stage 1: Zero Matching. If the matching score S is smaller than σ_1, we regard this stage as zero matching. Since there is almost no element of $\mathcal{G}_g$ observed in $\mathcal{G}_t$, the aim of agent at this stage is to expand its explored region and find elements in $\mathcal{G}_g$. Note this problem is similar to ON: before observing the goal, the agent needs to explore unknown regions without any matching between goal and scene. We can simply resort to scene graph-based ON method [44] at this stage, which navigates to frontiers with semantic relationships between the scene graph and goal as guidance.

However, different from ON where the goal is always an object node, in our universal goal-oriented navigation the goal may be a complicated graph. A graph may consist of several less related subgraph parts. For example, in a graph (table, chair, window, curtain), [table, chair] and [window, curtain] are two subgraphs which have strong internal correlation but weak interrelation. We empirically observe that locating a collection of multiple unrelated subgraphs in a scene at the same time is much more difficult than locating a single subgraph once at a time. Therefore, we decompose $\mathcal{G}_g$ into multiple internally correlated subgraphs with the guidance of LLM. For each subgraph of $\mathcal{G}_g$, we convert it to a text description, which is regarded as an object goal to call [44] for frontier selection. Finally, we select one frontier from the proposed ones by averaging the frontier scores and distances to the agent. Details about LLM-guided decomposition and score computation can be found in supplementary material. With this strategy, we not only utilize the information of the entire $\mathcal{G}_g$, but also eliminate ambiguity during frontier selection caused by unrelated subgraphs.

Stage 2: Partial Matching. With the exploration of agent, the elements of $\mathcal{G}_g$ will gradually be observed in $\mathcal{G}_t$ and thus S continues to increase. When S exceeds σ_1 (but still smaller than σ_2) and there is at least one anchor pair, we switch to partial matching stage. Here anchor pair means a pair of exactly matched nodes, i.e., two unconnected matched nodes in $\mathcal{M}_N$ or one matched edge in $\mathcal{M}_M$.

Note that we store the world coordinates of nodes in $\mathcal{G}_t$. If we also know the relative coordinates of nodes in $\mathcal{G}_g$, we can map the coordinates of $\mathcal{G}_t$ and $\mathcal{G}_g$ to bird's-eye view (BEV) and align the anchor pair of $\mathcal{G}_g$ to the one of $\mathcal{G}_t$. In this way, after alignment we can directly infer where the rest of $\mathcal{G}_g$ is, which provides the agent with clear exploration goal for each anchor pair. Luckily, although we do not have any coordinate information about the goal, at least we are aware of the relative spatial relationship between nodes in $\mathcal{G}_g$, such as a chair *on the left of* a table, a keyboard *in front of* a monitor. Inspired by this, we propose a coordinate projection strategy that preserves spatial relationships.

Given $\mathcal{G}_g$ without any coordinate information, we first project the central object node o to $(0, 0)$ as an initialization. To infer the projected coordinates of other nodes, we need to utilize the spatial relationship between coordinate-known nodes and coordinate-unknown nodes. Since at beginning we only have one coordinate-known node o, we start from it and traverse the goal graph with Depth First Search (DFS) to

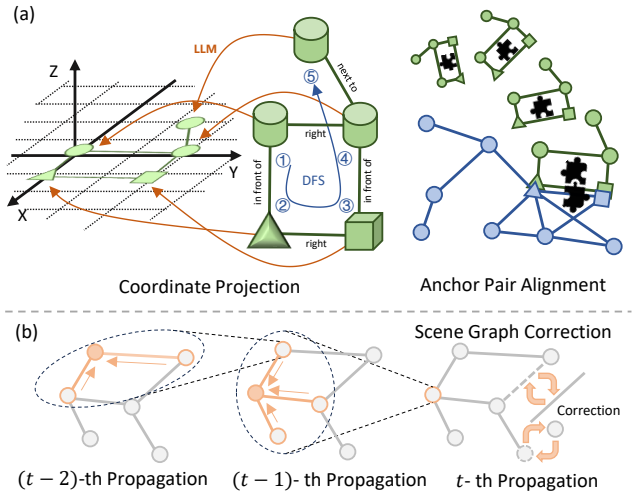

Figure 3. Illustration of approach. (a) Stage 2: coordinate projection and anchor pair alignment. (b) Stage 3: scene graph correction.

gradually infer the projected BEV coordinates of the whole graph. During the traversal, we focus on the edge connecting last node (coordinate-known) and current node (coordinate-unknown), which stores the spatial relationship. For each edge, we prompt LLM with: *[Current Node] is [Relationship] [Last Node], coordinate of [Last Node] is* (x, y)*, the X-axis and Y-axis are positive towards yourself and the right respectively. What is the coordinate of [Current Node]?* to project the current coordinate-unknown node to BEV.

With the projected $\mathcal{G}_t$ and $\mathcal{G}_g$, we conduct alignment for each anchor pair in order. Assume that the anchors are $\mathbf{v}_t^1, \mathbf{v}_t^2 \in \mathcal{G}_t$ and $\mathbf{v}_g^1, \mathbf{v}_g^2 \in \mathcal{G}_g$. $\mathbf{P} \in \mathbb{R}^{3\times3}$ is the 2D coordinate transfer matrix, consisting of scale $\mathbf{S}$, rotation $\mathbf{R}$ and translation $\mathbf{T}$, which is formulated as:

$$\mathbf{P} = \mathbf{S} \cdot \mathbf{R} \cdot \mathbf{T} = \begin{bmatrix} s\cos(\theta) & -s\sin(\theta) & t_x \\ s\sin(\theta) & s\cos(\theta) & t_y \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

Based on the anchor pair relationship, we establish the equation $\mathbf{v}_t^1 = \mathbf{P} \cdot \mathbf{v}_g^1$ and $\mathbf{v}_t^2 = \mathbf{P} \cdot \mathbf{v}_g^2$. Then we can solve the parameters t_x, t_y, θ, s in $\mathbf{P}$. With the coordinate transfer matrix $\mathbf{P}$, we project the rest nodes $\mathbf{v}_g \subset \mathcal{V}_g$ into the coordinate of $\mathcal{G}_t$ as $\mathbf{v}_t = \mathbf{P} \cdot \mathbf{v}_g$. Finally, the exploration goal for this anchor pair can be set as the center $\mathbf{c}^*$ of the smallest circumcircle of the projected nodes:

$$\mathbf{c}^* = \text{argmin}_{\mathbf{c}} \{r \mid \forall \mathbf{v} \in \mathbf{v}_t, \|\mathbf{v} - \mathbf{c}\| \leq r\} \tag{5}$$

which ensures that the distance from the exploration goal to the farthest node in $\mathbf{v}_t$ is minimal.

Stage 3: Perfect Matching. When the matching score S exceeds σ_2 and the central object o of $\mathcal{G}_g$ is matched ($o \in \mathcal{M}_N$), we switch the exploration policy to perfect matching stage. Since o is matched to an observed node in $\mathcal{G}_t$, we can simply have the agent move to this object without further exploration. However, there may be perceptual errors during scene graph construction. To ensure the matched o is correct, we propose a graph correction and goal verification pipeline to refine unreasonable structure in $\mathcal{G}_t$ as well as judge the confidence of goal in the process of approaching o.

We include nodes and edges of $\mathcal{G}_t$ within distance d of o into the correction scope, which is defined as a subgraph $\mathcal{G}_o = (\mathcal{V}_o, \mathcal{E}_o)$ with n nodes and m edges. As the agent approaches o, it continuously receives new RGB-D observation $\mathcal{I}^{(t)}$ and utilizes both visual observation and graph relationship to correct $\mathcal{G}_o$. Similar to graph convolution, we propagate information from neighbor nodes and edges and utilize LLM for information aggregation and updating:

$$\mathcal{V}_o^{(t+1)} = \text{LLM}(\mathbf{A} \cdot \mathcal{V}_o^{(t)}, \mathbf{M} \cdot \mathcal{E}_o^{(t)}, \text{VLM}(\mathcal{I}^{(t)})) \tag{6}$$

$$\mathcal{E}_o^{(t+1)} = \text{LLM}(\mathbf{M}^T \cdot \mathcal{V}_o^{(t)}, \mathbf{A}' \cdot \mathcal{E}_o^{(t)}, \text{VLM}(\mathcal{I}^{(t)})) \tag{7}$$

where $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{A}' \in \mathbb{R}^{m\times m}$ and $\mathbf{M} \in \mathbb{R}^{n\times m}$ are the adjacency matrix, edge adjacency matrix and incidence matrix of $\mathcal{G}_o$. $\mathcal{V}_o^{(t)}$ and $\mathcal{E}_o^{(t)}$ are the description of nodes and edges after t-th propagation. After several iterations, we prompt LLM to update unreasonable nodes and edges based on the aggregated $\mathcal{V}_o^{(t)}$ and $\mathcal{E}_o^{(t)}$. We detail the prompts in supplementary material.

To verify the goal, we consider several confidence items since entering stage 3 as follows:

$$C_t = N_t + M_t + S_t - \lambda D_t \tag{8}$$

where N_t, M_t, S_t are the proportions of corrected nodes and edges, matched keypoints using LightGlue [22] (for IIN only) and graph matching score at time t. D_t is the path length since stage 3. If C_t exceeds a threshold C_{thr} within t steps, o will be verified. Otherwise o will be excluded.

3.4. Robust Blacklist Mechanism

Since our graph matching method always outputs matched result which maximizes the similarity score, if one matching between $\mathcal{G}_t$ and $\mathcal{G}_g$ finally fails to find the goal, relevant nodes and edges in $\mathcal{G}_t$ should be marked to avoid repeated matching. To this end, we present a blacklist mechanism to record the unsuccessful matching.

Blacklist is initialized as empty. The nodes and edges in blacklist will not be considered in our graph matching method. Two cases will extend the blacklist: (1) all anchor pairs of stage 2 in a single matching fails to enter stage 3. In this case, the nodes and edges in these anchor pairs will be appended to blacklist; (2) the goal verification of stage 3 fails. This will move all matched pairs in $\mathcal{M}_N$ and $\mathcal{M}_E$ to blacklist. Besides, if any node or edge is refined during scene graph correction of stage 3, these nodes and edges along with their connected ones will be removed from blacklist.

Table 1. Results of Object-goal navigation, Instance-image-goal navigation and Text-goal navigation on MP3D, HM3D and RoboTHOR. We compare the SR and SPL of state-of-the-art methods in different settings. Universal goal-oriented navigation methods are colored in gray.

Method	Training-Free	Universal	ObjNav						InsINav		TextNav	
			MP3D		HM3D		RoboTHOR		HM3D		HM3D	
			SR	SPL	SR	SPL	SR	SPL	SR	SPL	SR	SPL
SemEXP [7]	×	×	36.0	14.4	–	–	–	–	–	–	–	–
ZSON [27]	×	×	15.3	4.8	25.5	12.6	–	–	–	–	–	–
OVRL-v2 [43]	×	×	–	–	64.7	28.1	–	–	–	–	–	–
Krantz et al. [15]	×	×	–	–	–	–	–	–	8.3	3.5	–	–
OVRL-v2-IIN [43]	×	×	–	–	–	–	–	–	24.8	11.8	–	–
IEVE [19]	×	×	–	–	–	–	–	–	70.2	25.2	–	–
PSL [37]	×	✓	–	–	42.4	19.2	–	–	23.0	11.4	16.5	7.5
GOAT [6]	×	✓	–	–	50.6	24.1	–	–	37.4	16.1	17.0	8.8
ESC [49]	✓	×	28.7	14.2	39.2	22.3	38.1	22.2	–	–	–	–
OpenFMNav [17]	✓	×	37.2	15.7	52.5	24.1	44.1	23.3	–	–	–	–
VLFM [45]	✓	×	36.2	15.9	52.4	**30.3**	42.3	23.0	–	–	–	–
SG-Nav [44]	✓	×	40.2	16.0	54.0	24.9	47.5	24.0	–	–	–	–
Mod-IIN [16]	✓	×	–	–	–	–	–	–	56.1	23.3	–	–
UniGoal	✓	✓	**41.0**	**16.4**	**54.5**	25.1	**48.0**	**24.2**	**60.2**	**23.7**	**20.2**	**11.4**

4. Experiments

In this section, we conduct extensive experiments to validate the effectiveness of UniGoal. We first describe experimental settings. Then we compare UniGoal with state-of-the-art methods and ablate each component in UniGoal. Finally we demonstrate some qualitative results.

4.1. Benchmarks and Implementation Details

Datasets: We evaluate UniGoal on object-goal, instance-image-goal and text-goal navigation. For ON, we conduct experiments on the widely used Matterport3D (MP3D [4]), Habitat-Matterport 3D (HM3D [30]) and RoboTHOR [10] following the setting of SG-Nav [44]. For IIN and TN, we compare with other methods on HM3D, following Mod-IIN [16] and InstanceNav [37] respectively.

Evaluation Metrics: We report *success rate (SR)* and *success rate weighted by path length (SPL)*. SR represents the proportion of successful navigation episodes, while SPL measures how close the path is to the optimal path.

Compared Methods: We compare with previous state-of-the-art methods on the studied three tasks. For ON, we compare with the supervised methods SemEXP [7], ZSON [27], OVRL-v2 [43], and zero-shot methods ESC [49], OpenFMNav [17], VLFM [45], SG-Nav [44]. For IIN, we compare with the supervised methods Krantz et al. [15], OVRL-v2 (implementation from [19]), IEVE [19], and zero-shot methods Mod-IIN [16]. For TN, since currently there is no zero-shot method available, we compare with the supervised methods PSL [37] and GOAT [6]. PSL and GOAT are also universal methods, for which we also compare with them on other two tasks.

Implementation Details: We set up our agent in Habitat Simulator [35, 38]. We deploy LLaMA-2-7B [39] as the LLM and LLaVA-v1.6-Mistral-7B [23] as the VLM throughout the text. We adopt CLIP [28] text encoder to extract the embedding of nodes and edges during graph matching. Hyperparameters are detailed in appendix.

4.2. Comparison with State-of-the-art

We compare UniGoal with the state-of-the-art goal-oriented navigation methods of different setting, including supervised, zero-shot and universal methods on three studied tasks in Table 1. On zero-shot ON and IIN, UniGoal surpasses state-of-the-art methods SG-Nav and Mod-IIN by 0.8% and 4.1% respectively. We achieve higher performance even compared with some supervised methods, like SemEXP and ZSON. Note that the improvement of UniGoal on IIN is more significant than ON. This is because the goal in ON is a single object node, which cannot fully exploit the potential of UniGoal on graph matching and reasoning. Moreover, UniGoal also achieves state-of-the-art performance among universal goal-oriented methods, even outperforming supervised methods like PSL and GOAT with $+3.9/1.0$ lead on ON, $+22.8/7.6$ lead on IIN and $+3.2/2.6$ lead on TN.

It is worth noting that for ON, our goal graph degenerates to a single node and thus some modules in UniGoal will not work, such as the goal graph decomposition in stage 1 and anchor pair alignment in stage 2. In this case, UniGoal still outperforms the scene graph-based zero-shot method SG-Nav on all benchmarks, which validates the effectiveness of graph correction and goal verification method in stage 3.

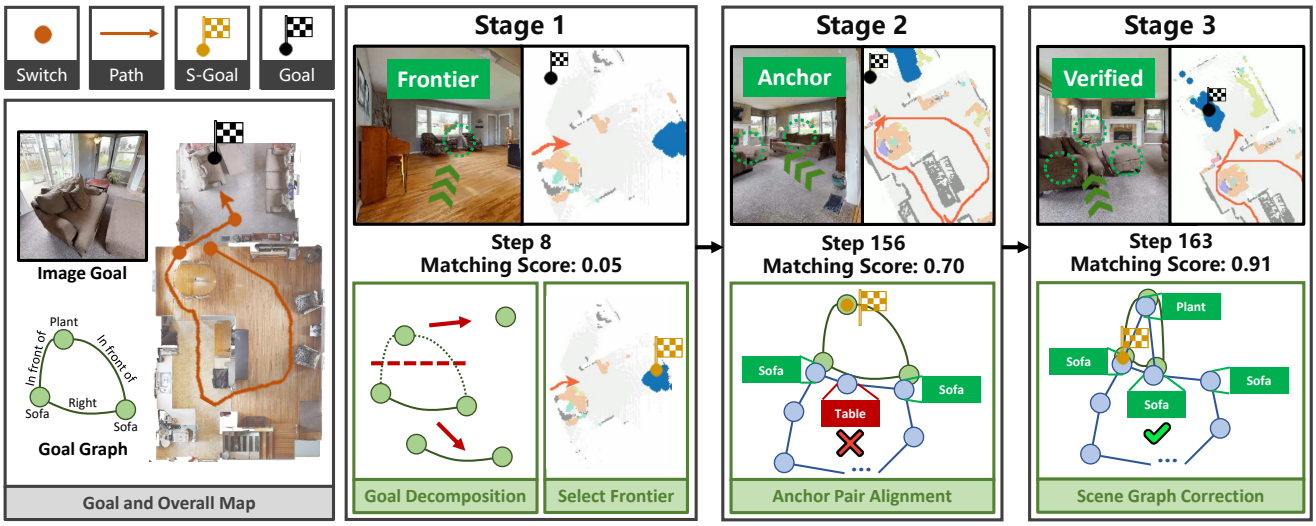

Figure 4. Demonstration of the decision process of UniGoal. Here 'Switch' means the point when stage is changing. 'S-Goal' means the long-term goal predicted in each stage.

Table 2. Effect of pipeline design in UniGoal on HM3D (IIN) benchmark.

Method	SR	SPL
Simplify graph matching	54.9	20.7
Remove blacklist mechanism	50.6	17.3
Simplify multi-stage exploration policy	59.0	23.2
Full Approach	**60.2**	**23.7**

Table 3. Effect of the submodules in each stage during multi-stage scene exploration on HM3D (IIN) benchmark.

Method	Stage	SR	SPL
Replace stage 1 with FBE	1	55.1	20.8
Remove $\mathcal{G}_g$ decomposition	1	59.2	22.6
Remove frontier selection	1	57.4	22.0
Simplify coordinate projection	2	59.1	22.7
Remove anchor pair alignment	2	58.9	22.6
Remove $\mathcal{G}_t$ correction	3	59.5	23.5
Remove goal verification	3	58.2	22.4
Full Approach	–	**60.2**	**23.7**

4.3. Ablation Study

We conduct ablation experiments on HM3D to validate the effectiveness of each part in UniGoal. We report performance of the ablated versions on the representative IIN task.

Pipeline Design. In Table 2, we ablate the graph matching method and multi-stage explore policy. We first simplify graph matching by removing the score computation. In this way, once anchor pair is matched, the agent will enter stage 2. Similarly, the agent will enter stage 3 once the central object o of $\mathcal{G}_g$ is matched. It is shown that the agent cannot switch the exploration strategy at the optimal time without a judgment of matching degree, leading to more failed cases. Then we remove the blacklist mechanism and observe a significant performance degradation. This is due to repeated matching on some failed nodes and edges, which makes the agent get stuck. Finally we simplify the multi-stage exploration policy by removing stage 2. The agent will directly switch from unknown region exploration to goal approaching according to the matching result. It is shown that the performance drops significantly when applying a simpler exploration policy.

Effect of Each Exploration Stage. As shown in Table 3, we conduct ablation studies on each exploration stage. For stage 1, we first replace the whole stage with a simple FBE strategy. Then we remove the goal graph decomposition and prompt the LLM with whole graph. We also remove frontier selection by making LLM predict a point location of the goal rather than scoring each frontier. The results in the first three rows show that each component in stage 1 is effective. For stage 2, we first simplify the LLM-based coordinate projection method to a random guess of the 2D coordinate. We then remove the anchor pair alignment method by directly making LLM predict the goal location based on the BEV graphs. The experimental results in the fourth and fifth rows validate the effectiveness of inferring the goal location with structure overlap between graphs. For stage 3, we remove the scene graph correction and goal verification in turn and report the results in the sixth and seventh rows. It is shown that both method works well to improve our final performance, which means correcting the scene graph as well as verifying the goal graph are essential for robust navigation.

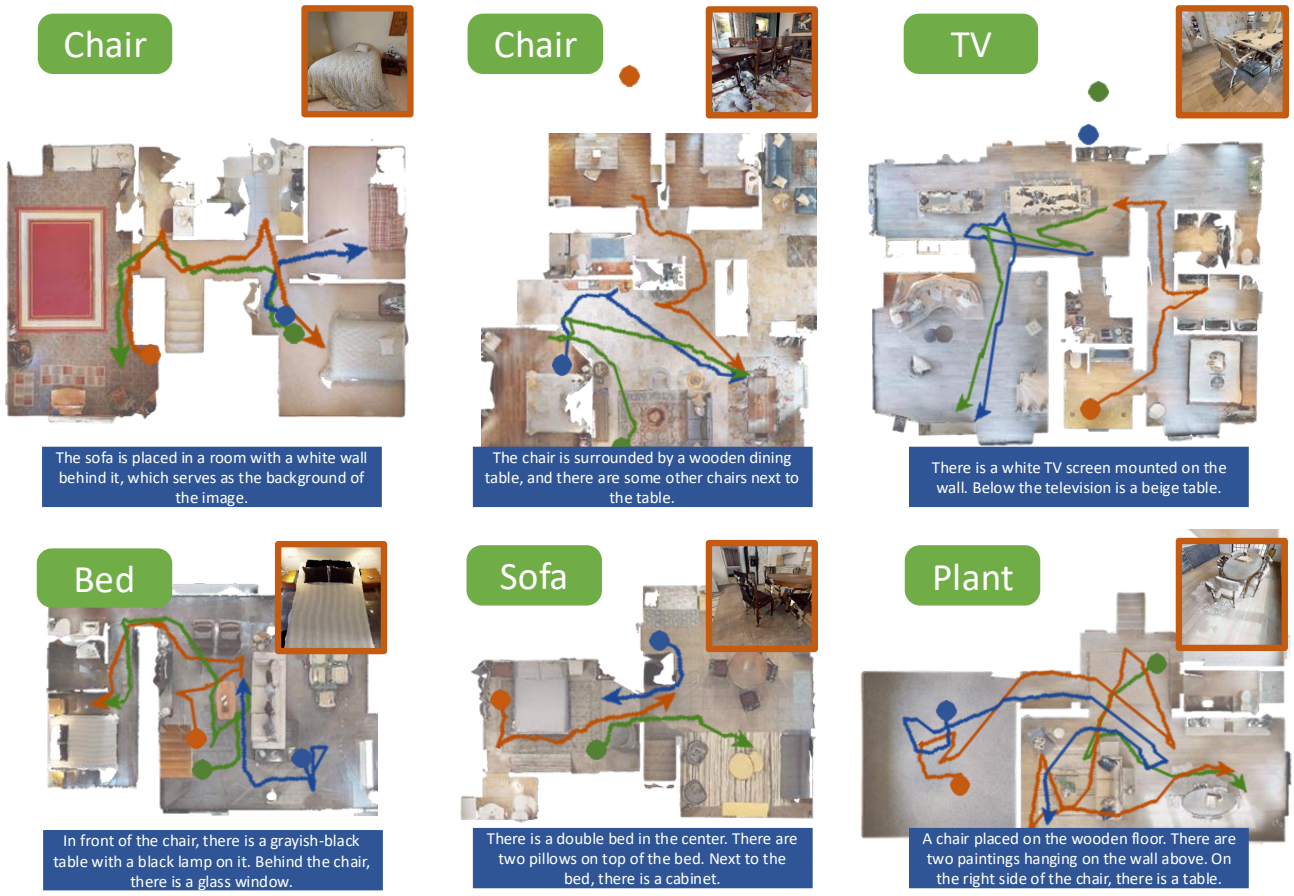

Figure 5. Visualization of the navigation path. We visualize ON (Green), IIN (Orange) and TN (Blue) path for several scenes. UniGoal successfully navigates to the target given different types of goal and diverse environments.

4.4. Qualitative Results

We provide some qualitative results of UniGoal for better understanding. We first demonstrates the decision process of UniGoal in Figure 4. It is shown that UniGoal gradually increases the matching score by graph-based exploration. Our multi-stage policy outputs reasonable stage-wise long-term goal with current observation of the environment. We also visualize the navigation path of UniGoal on the three studied tasks on 9 scenes of HM3D in Figure 5. It is shown that UniGoal can effectively handle all tasks with a single model, and generates efficient trajectory to reach the goal.

5. Conclusion

In this paper, we have presented UniGoal, a universal zero-shot goal-oriented navigation framework which can handle object-goal navigation, instance-image-goal navigation and text-goal navigation in a single model without training or finetuning. Since different types of goal usually requires totally different goal representation and LLM inference pipeline for zero-shot navigation, it is challenging to unify them with a single framework. To solve this problem, we convert the agent's observation into a scene graph and propose a uniform graph-based goal representation. In this way, the scene and goal are represented consistently, based on which we design a graph matching and matching-guided multi-stage scene exploration policy to make LLM fully exploit the correlation between scene and goal. Under different overlap, we propose different strategies to locate the goal position in the scene graph. Besides, we maintain a blacklist that records unsuccessful matching to avoid repeated exploration. Experimental results on three widely used datasets validate the effectiveness of UniGoal. We further deploy UniGoal on real-world robotic platform to demonstrate its strong generalization ability and application value.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 62125603, Grant 62336004, Grant 62321005, and Grant 62441616, and in part by the Beijing Natural Science Foundation under Grant No. L247009.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2, 3

[2] Iro Armeni, Zhi-Yang He, JunYoung Gwak, Amir R. Zamir, Martin Fischer, Jitendra Malik, and Silvio Savarese. 3d scene graph: A structure for unified semantics, 3d space, and camera, 2019. 2

[3] Wenzhe Cai, Siyuan Huang, Guangran Cheng, Yuxing Long, Peng Gao, Changyin Sun, and Hao Dong. Bridging zero-shot object navigation and foundation models through pixel-guided navigation skill. In *ICRA*, pages 5228–5234. IEEE, 2024. 2

[4] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *3DV*, 2017. 2, 6

[5] Haonan Chang, Kowndinya Boyalakuntla, Shiyang Lu, Siwei Cai, Eric Pu Jing, Shreesh Keskar, Shijie Geng, Adeeb Abbas, Lifeng Zhou, Kostas Bekris, and Abdeslam Boularias. Context-aware entity grounding with open-vocabulary 3d scene graphs. In *CoRL*, pages 1950–1974. PMLR, 2023. 2

[6] Matthew Chang, Theophile Gervet, Mukul Khanna, Sriram Yenamandra, Dhruv Shah, So Yeon Min, Kavit Shah, Chris Paxton, Saurabh Gupta, Dhruv Batra, et al. Goat: Go to any thing. *arXiv preprint arXiv:2311.06430*, 2023. 1, 2, 6

[7] Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Russ R Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. In *NeurIPS*, pages 4247–4258, 2020. 1, 2, 3, 6

[8] Boyuan Chen, Fei Xia, Brian Ichter, Kanishka Rao, Keerthana Gopalakrishnan, Michael S Ryoo, Austin Stone, and Daniel Kappler. Open-vocabulary queryable scene representations for real world planning. In *ICRA*, pages 11509–11522. IEEE, 2023. 2

[9] Jiaqi Chen, Bingqian Lin, Ran Xu, Zhenhua Chai, Xiaodan Liang, and Kwan-Yee K. Wong. Mapgpt: Map-guided prompting with adaptive path planning for vision-and-language navigation. In *ACL*, 2024. 2

[10] Matt Deitke, Winson Han, Alvaro Herrasti, Aniruddha Kembhavi, Eric Kolve, Roozbeh Mottaghi, Jordi Salvador, Dustin Schwenk, Eli VanderBilt, Matthew Wallingford, Luca Weihs, Mark Yatskar, and Ali Farhadi. Robothor: An open simulation-to-real embodied ai platform. In *CVPR*, 2020. 2, 6

[11] Yilun Du, Chuang Gan, and Phillip Isola. Curious representation learning for embodied intelligence. In *CVPR*, 2021. 2

[12] Samir Yitzhak Gadre, Mitchell Wortsman, Gabriel Ilharco, Ludwig Schmidt, and Shuran Song. Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation. In *CVPR*, pages 23171–23181, 2023. 2

[13] Qiao Gu, Alihusein Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, et al. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. *arXiv preprint arXiv:2309.16650*, 2023. 2

[14] Ue-Hwan Kim, Jin-Man Park, Taek-Jin Song, and Jong-Hwan Kim. 3-d scene graph: A sparse and semantic representation of physical environments for intelligent agents. *TCybern*, 50 (12):4921–4933, 2019. 2

[15] Jacob Krantz, Stefan Lee, Jitendra Malik, Dhruv Batra, and Devendra Singh Chaplot. Instance-specific image goal navigation: Training embodied agents to find object instances. *arXiv preprint arXiv:2211.15876*, 2022. 1, 3, 6

[16] Jacob Krantz, Theophile Gervet, Karmesh Yadav, Austin Wang, Chris Paxton, Roozbeh Mottaghi, Dhruv Batra, Jitendra Malik, Stefan Lee, and Devendra Singh Chaplot. Navigating to objects specified by images. In *ICCV*, pages 10916–10925, 2023. 1, 2, 6

[17] Yuxuan Kuang, Hai Lin, and Meng Jiang. Openfmnav: Towards open-set zero-shot object navigation via vision-language foundation models. *arXiv preprint arXiv:2402.10670*, 2024. 6

[18] Obin Kwon, Jeongho Park, and Songhwai Oh. Renderable neural radiance map for visual navigation. In *CVPR*, pages 9099–9108, 2023. 2

[19] Xiaohan Lei, Min Wang, Wengang Zhou, Li Li, and Houqiang Li. Instance-aware exploration-verification-exploitation for instance imagegoal navigation. In *CVPR*, pages 16329–16339, 2024. 6

[20] Dingbang Li, Wenzhou Chen, and Xin Lin. Tina: Think, interaction, and action framework for zero-shot vision language navigation. *arXiv preprint arXiv:2403.08833*, 2024. 2

[21] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *CVPR*, pages 10965–10975, 2022. 1

[22] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local Feature Matching at Light Speed. In *ICCV*, 2023. 5

[23] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. 6, 1

[24] Rui Liu, Xiaohan Wang, Wenguan Wang, and Yi Yang. Bird's-eye-view scene graph for vision-language navigation. In *ICCV*, pages 10968–10980, 2023. 2

[25] Yuxing Long, Wenzhe Cai, Hongcheng Wang, Guanqi Zhan, and Hao Dong. Instructnav: Zero-shot system for generic instruction navigation in unexplored environment. *arXiv preprint arXiv:2406.04882*, 2024. 2

[26] Yuxing Long, Xiaoqi Li, Wenzhe Cai, and Hao Dong. Discuss before moving: Visual language navigation via multi-expert discussions. In *ICRA*, pages 17380–17387. IEEE, 2024. 2

[27] Arjun Majumdar, Gunjan Aggarwal, Bhavika Devnani, Judy Hoffman, and Dhruv Batra. Zson: Zero-shot object-goal navigation using multimodal goal embeddings. *NeurIPS*, 35: 32340–32352, 2022. 6

[28] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR, 2021. 2, 6

[29] Abhinav Rajvanshi, Karan Sikka, Xiao Lin, Bhoram Lee, Han-Pang Chiu, and Alvaro Velasquez. Saynav: Grounding large language models for dynamic planning to navigation in new environments. In *ICAPS*, pages 464–474, 2024. 2

[30] Santhosh K. Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alex Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X. Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai, 2021. 2, 6

[31] Santhosh Kumar Ramakrishnan, Devendra Singh Chaplot, Ziad Al-Halah, Jitendra Malik, and Kristen Grauman. Poni: Potential functions for objectgoal navigation with interaction-free learning. In *CVPR*, pages 18890–18900, 2022. 2

[32] Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian Reid, and Niko Suenderhauf. Sayplan: Grounding large language models using 3d scene graphs for scalable task planning. In *CoRL*, 2023. 2

[33] Antoni Rosinol, Arjun Gupta, Marcus Abate, Jingnan Shi, and Luca Carlone. 3d dynamic scene graphs: Actionable spatial perception with places, objects, and humans, 2020. 2

[34] Alberto Sanfeliu and King-Sun Fu. A distance measure between attributed relational graphs for pattern recognition. *TSMC*, (3):353–362, 1983. 4

[35] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *ICCV*, pages 9339–9347, 2019. 6

[36] James A Sethian. Fast marching methods. *SIAM review*, 41 (2):199–235, 1999. 4

[37] Xander Sun, Louis Lau, Hoyard Zhi, Ronghe Qiu, and Junwei Liang. Prioritized semantic learning for zero-shot instance navigation. In *ECCV*, 2024. 1, 2, 3, 6

[38] Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Singh Chaplot, Oleksandr Maksymets, et al. Habitat 2.0: Training home assistants to rearrange their habitat. *NeurIPS*, 34:251–266, 2021. 6

[39] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 2, 3, 6

[40] Johanna Wald, Helisa Dhamo, Nassir Navab, and Federico Tombari. Learning 3d semantic scene graphs from 3d indoor reconstructions, 2020. 2

[41] Erik Wijmans, Abhishek Kadian, Ari Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra. Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames. *arXiv preprint arXiv:1911.00357*, 2019. 2

[42] Pengying Wu, Yao Mu, Bingxian Wu, Yi Hou, Ji Ma, Shanghang Zhang, and Chang Liu. Voronav: Voronoi-based zero-shot object navigation with large language model. *arXiv preprint arXiv:2401.02695*, 2024. 2

[43] Karmesh Yadav, Arjun Majumdar, Ram Ramrakhya, Naoki Yokoyama, Alexei Baevski, Zsolt Kira, Oleksandr Maksymets, and Dhruv Batra. Ovrl-v2: A simple state-of-art baseline for imagenav and objectnav. *arXiv preprint arXiv:2303.07798*, 2023. 6

[44] Hang Yin, Xiuwei Xu, Zhenyu Wu, Jie Zhou, and Jiwen Lu. Sg-nav: Online 3d scene graph prompting for llm-based zero-shot object navigation. In *NeurIPS*, 2024. 1, 2, 3, 4, 6

[45] Naoki Yokoyama, Sehoon Ha, Dhruv Batra, Jiuguang Wang, and Bernadette Bucher. Vlfm: Vision-language frontier maps for zero-shot semantic navigation. In *ICRA*, 2024. 2, 6

[46] Bangguo Yu, Hamidreza Kasaei, and Ming Cao. L3mvn: Leveraging large language models for visual target navigation. In *IROS*, pages 3554–3560. IEEE, 2023. 2

[47] Zhaohuan Zhan, Lisha Yu, Sijie Yu, and Guang Tan. Mc-gpt: Empowering vision-and-language navigation with memory map and reasoning chains. *arXiv preprint arXiv:2405.10620*, 2024. 2

[48] Gengze Zhou, Yicong Hong, and Qi Wu. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. In *AAAI*, pages 7641–7649, 2024. 2

[49] Kaiwen Zhou, Kaizhi Zheng, Connor Pryor, Yilin Shen, Hongxia Jin, Lise Getoor, and Xin Eric Wang. Esc: Exploration with soft commonsense constraints for zero-shot object navigation. In *ICML*, pages 42829–42842. PMLR, 2023. 1, 2, 6