

# UniGoal: Towards Universal Zero-shot Goal-oriented Navigation

## Supplementary Material

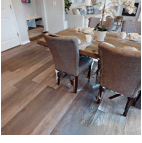
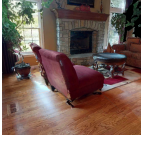
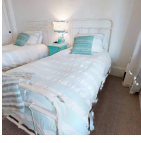
ON	Plant	Chair	Toilet
IIN			
	Chair	Sofa	Bed
TN	The <b>toilet</b> in this image is white, surrounded by a white door, beige tiles on the walls and floor.	The <b>bed</b> has white bedsheets. The bedroom has a double bed, two pillows and blankets, a chair and a table.	The <b>chair</b> is yellow and covered with red floral patterns. There is a wooden dining table in the upper left corner.

Table 4. Illustration of goal in each task, with central objects colored in red.

### A. Overview

This supplementary material is organized as follows:

- Section B provides the details of the three studied tasks.
- Section C provides the overall pipeline of UniGoal in algorithm for better understanding.
- Section D provides details on the approach.
- Section E details the prompts for LLM.

### B. Definition of Each Task

We provide samples of the goal in each task in Table 4 for better understanding. The goal for ON is a category in text format. The goal for IIN is an image with  $o$  located at the center. For TN, the goal is a description about  $o$ , such as its relationship with other relevant objects in the scene.

### C. Pipeline of UniGoal

We provide the algorithm diagram of UniGoal in Algorithm 1. One gray box represents a stage.

### D. Details of Approach

#### D.1. Goal Graph Construction

For Object-goal, we simply construct a graph with only one node and no edge, where the content of the node is the category of goal. For Instance-image-goal, we adopt Grounded-

#### Algorithm 1 Overall Pipeline of UniGoal

**Require:** Goal  $g$ , Observation  $\mathcal{I}$

**Ensure:** Goal Position  $(x, y)$

$\mathcal{G}_t \leftarrow \text{NewSceneGraph}()$

$\mathcal{G}_g \leftarrow \text{ConstructGoalGraph}(g)$

$\mathcal{B} \leftarrow \text{NewBlackList}()$

**while** True **do**

$\mathcal{G}_t \leftarrow \text{UpdateSceneGraph}(\mathcal{G}_t, \mathcal{I})$

$S, \mathcal{M}_N, \mathcal{M}_E \leftarrow \text{GraphMatching}(\mathcal{G}_t, \mathcal{G}_g)$

Stage2  $\leftarrow (\sigma_1 \leq S < \sigma_2 \text{ and } |\mathcal{M}_N| \geq 2)$

Stage3  $\leftarrow (S > \sigma_2 \text{ and } o \in \mathcal{M}_N)$

$i \leftarrow 0$

===== Stage 1 =====

**if not** (Stage2 or Stage3) **then**

$\mathcal{G}_g^{sub} \leftarrow \text{GraphDecomposition}(\mathcal{G}_g)$

$(x, y) \leftarrow \text{SearchFrontier}(\mathcal{G}_g^{sub}, \mathcal{G}_t \setminus \mathcal{B})$

Go to  $(x, y)$

**end if**

===== Stage 2 =====

**if** Stage2 **then**

$\mathcal{G}'_t, \mathcal{G}'_g \leftarrow \text{CoordinateProjection}(\mathcal{G}_t \setminus \mathcal{B}, \mathcal{G}_g)$

$\mathbb{P} \leftarrow \text{AnchorPairAlign}(\mathcal{M}_N, \mathcal{M}_E, \mathcal{G}'_t, \mathcal{G}'_g)$

#  $\mathbb{P} = \{(x_i, y_i) | i = 1, 2, \dots, n\}$

**if**  $1 \leq i \leq n$  **then**

Go to  $(x_i, y_i)$

$i \leftarrow i + 1$

**else**

$\mathcal{B} \leftarrow \text{UpdateBlackList}(\mathcal{M}_N, \mathcal{M}_E, \mathcal{B})$

$i \leftarrow 0$

**end if**

**end if**

===== Stage 3 =====

**if** Stage3 **then**

$\mathcal{G}_t \leftarrow \text{SceneGraphCorrection}(\mathcal{G}_t, \mathcal{I})$

**if** GoalVerification( $\mathcal{G}_g, \mathcal{I}$ ) **then**

Go to  $(x, y)$  # Navigation Stop

**else**

$\mathcal{B} \leftarrow \text{UpdateBlackList}(\mathcal{M}_N, \mathcal{M}_E, \mathcal{B})$

**end if**

**end if**

**end while**

SAM [21] to identify all the objects in the image, and then we prompt VLM [23] to identify the relationships between objects to construct edges. For Text-goal, we prompt LLM to first identify all objects in the description, and then generate relationships between objects.

## D.2. Graph Embedding

To embed nodes ( $\mathcal{V}_t$  and  $\mathcal{V}_g$ ), the embedding function is implemented as  $\text{Embed}(v) = \text{concat}(\text{CLIP}(v), \text{Degree}(v))$ , where  $\text{CLIP}(\cdot)$  is CLIP [28] text encoder,  $\text{Degree}(\cdot)$  is the degree of a node. To embed edges ( $\mathcal{E}_t$  and  $\mathcal{E}_g$ ), the embedding function is implemented as  $\text{Embed}(e) = \text{CLIP}(e)$ .

## D.3. Zero Matching

We detail the LLM-guided graph decomposition and frontier score computation used in zero-matching stage in this subsection.

**LLM-guided Graph Decomposition.** We decompose the  $\mathcal{G}_g$  into several subgraphs  $\mathcal{G}_g^i$  by dividing  $\mathcal{V}_g$  and assign the edges into each subgraph. The node sets  $\mathcal{V}_g^i$  of all subgraphs comprise a division of  $\mathcal{V}_g$ , i.e., they satisfy:

$$\bigcup_i \mathcal{V}_g^i = \mathcal{V}_g$$

$$\mathcal{V}_g^i \cap \mathcal{V}_g^j = \emptyset \quad \forall i \neq j$$

We prompt LLM to conduct the division of  $\mathcal{V}_g$ . Then we assign all the edges whose connected two nodes are in the same  $\mathcal{V}_g^i$  into the subgraph  $\mathcal{G}_g^i$ . The edges whose connected two nodes are not in a same  $\mathcal{V}_g^i$  will be excluded during decomposition.

**Frontier Scoring.** Following SG-Nav, we prompt LLM with chain-of-thought (CoT) to predict the most likely position of  $\mathcal{V}_g^i$  and score the frontiers according to their distance to the most likely position and their distance to the agent. Then we select the center of frontier with maximum score as the long-term goal.

## D.4. Hyperparameters

The maximum navigation step number  $T$  is set as:  $T = 500$  for ON,  $T = 1000$  for IIN and TN. The distance of success condition  $r$  is set as:  $r = 1.6\text{m}$  for ON,  $r = 1.0\text{m}$  for IIN and TN. The threshold of similarity for matching pairs of nodes and edges  $\tau$  is set as:  $\tau = 0.9$ . The matching score thresholds are set as:  $\sigma_1 = 0.5, \sigma_2 = 0.9$ .

## E. Prompts

We provide all prompts used in UniGoal.

### Goal Construction for IIN:

You are an AI assistant that can infer relationships between objects. You need to guess the spatial relationship between {object1} and {object2} in the {image}. Answer relationship with one word or phrase.

where {object1}, {object2} and {image} will be replaced by the two objects and the goal image.

### Goal Construction for TN:

You are an AI assistant that can identify objects and relationships from a description. You need to list the objects and relationships in {text} in following format:

```
{
  'nodes': [{ 'id': 'book' }, { 'id': 'table' }],
  'edges': [{ 'source': 'book',
               'target': 'table', 'type': 'on' },]
```

where {text} will be replaced by the goal text.

### Goal Decomposition:

You are an AI assistant with commonsense. You need to divide the following objects {} into subsets based on correlation, with objects within each subset strongly correlated and objects within different subsets not strongly correlated.

Your response should be in two-dimensional array format:

```
[[object1, object2, ..., objectn],
 [...], ..., [...]]
```

where {} will be replaced by the objects in  $\mathcal{V}_g$ .

### Frontier Scoring:

You are an AI assistant with commonsense. You need to predict the most likely distance between the {object} and the {subgraph}. Answer a distance number in meter.

where {object} and {subgraph} will be replaced by each object in  $\mathcal{G}_t$  and the description of decomposed  $\mathcal{G}_g$ .

### Scene Graph Correction:

You are an AI assistant with commonsense and strong ability to give a more detailed description of a node or edge in an indoor scene graph.

Now give a more detailed description of {} based on the graph {graph} and the newly observed image {image} in order to identify possible errors in the scene graph.

where {}, {graph} and {image} will be replaced by a node or edge, the local graph  $\mathbf{A} \cdot \mathcal{V}_o^{(t)}, \mathbf{M} \cdot \mathcal{E}_o^{(t)}$  or  $\mathbf{M}^T \cdot \mathcal{V}_o^{(t)}, \mathbf{A}' \cdot \mathcal{E}_o^{(t)}$  and the description of the newly observed image.