

Mapping High-level Semantic Regions in Indoor Environments without Object Recognition

Roberto Bigazzi¹, Lorenzo Baraldi¹, Shreyas Kousik², Rita Cucchiara¹, and Marco Pavone³

Abstract—Robots require a semantic understanding of their surroundings to operate in an efficient and explainable way in human environments. In the literature, there has been an extensive focus on object labeling and exhaustive scene graph generation; less effort has been focused on the task of purely identifying and mapping large semantic regions. The present work proposes a method for semantic region mapping via embodied navigation in indoor environments, generating a high-level representation of the knowledge of the agent. To enable region identification, the method uses a vision-to-language model to provide scene information for mapping. By projecting egocentric scene understanding into the global frame, the proposed method generates a semantic map as a distribution over possible region labels at each location. This mapping procedure is paired with a trained navigation policy to enable autonomous map generation. The proposed method significantly outperforms a variety of baselines, including an object-based system and a pretrained scene classifier, in experiments in a photorealistic simulator.

I. INTRODUCTION

A critical ingredient in robot autonomy is *semantic region mapping*, or the capability to recognize high-level semantics of a robot’s surroundings while creating a spatial map. In this work, we aim to recognize semantic areas at the region-level of an indoor environment such as a house or an office; *e.g.*, understanding if the robot is in a bedroom, a living room, *etc.* This task, which we call Indoor Semantic Region Mapping (ISRM), is often addressed by leveraging object detection to classify a region [1]–[5]. However, such an approach may fail when an object appears in multiple regions (*e.g.* a fridge in a garage), when regions are not easily delineated (*e.g.* an open kitchen attached to a living room), or when regions do not contain many objects (*e.g.* a hallway). Instead, we explore how to endow robots with a semantic recognition of their surroundings by processing sensor inputs holistically.

We note that semantic region recognition could be considered an instance of the well-defined Computer Vision task of Scene Recognition. However, directly adapting methods suited to this task is impractical because scene recognition methods are trained on large-scale offline datasets of everyday pictures, which differ significantly from the observations taken from the point of view of a robotic agent [6]. Furthermore, the category labels in scene-specific datasets

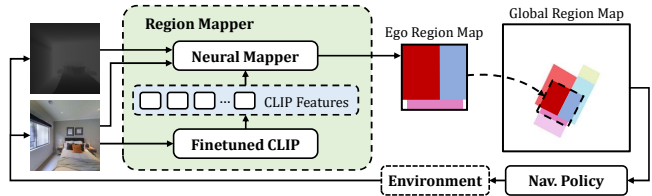


Fig. 1. We address the task of indoor semantic region mapping for Embodied Navigation, which requires an agent to build a global semantic understanding of the large-scale regions in an environment. Our approach conditions a learned neural mapper with visual features extracted using a general region classifier to produce a geometric region map of the environment.

are usually very specific and do not contain samples with occlusion or multiple categories.

To enable solving ISRM without relying on object recognition, we design a region classification module powered by CLIP [7], a finetuned, general image-text model. As CLIP has been trained on a large-scale image-text pair dataset containing natural images, it is generalizable, but it also shows poor performance when dealing with photorealistic indoor images taken from a simulator. Therefore, we devise a finetuning strategy with a multi-modal variant of the supervised contrastive loss [8], which allows us to obtain an effective region classifier, even with scarce training data. Further, we design a region mapping architecture that can predict egocentric and global occupancy and region maps, and a navigation module that can exploit the learned region semantics. We perform an extensive experimental analysis to compare different approaches to tackle the ISRM task. Overall, we see this research as a basic step for the next generation of personalized robotics autonomous systems, equipped with vision abilities for moving in complex environments and reasoning at a semantic level to make them suitable for new forms of interaction and cooperation with humans.

Contributions. In summary, we make three contributions. First, we propose a region classification module that generates grounded language-visual features, suitable for downstream mapping, by using an RGB-D camera and a finetuned CLIP model. Second, we propose an approach to fusing our region classification features into a global grid map by leveraging an exploration-focused navigation policy [9]. Third, we provide an extensive set of ablations and baselines to understand the utility and capability of our method, and find that it outperforms an object-based mapping system. That is, we show that grounding high-level region labels conveys an advantage in robotic semantic mapping.

¹Roberto Bigazzi, Lorenzo Baraldi, and Rita Cucchiara are with the Department of Engineering “Enzo Ferrari”, University of Modena and Reggio Emilia, Italy {firstname.lastname}@unimore.it

²Shreyas Kousik is with the George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, USA shreyas.kousik@me.gatech.edu

³Marco Pavone is with the Department of Aeronautics & Astronautics, Stanford University, USA pavone@stanford.edu

II. RELATED WORK

We now review relevant work on learning-based mapping, semantic mapping, and scene recognition.

Learning-Based Mapping. A variety of approaches leverage learning to overcome shortcomings in classical SLAM (Simultaneous Localization and Mapping) (*c.f.*, [10]–[12]). For example, one can learn to generate classical SLAM-style maps [13], [14], topological maps [15], [16], multi-task deep memory representations [17], [18], and inferences over unseen regions [19]. In this work, we propose a learning-based approach to mapping semantic regions.

Semantic Mapping. Semantic mapping approaches can be grouped into two main categories based on generating either low-level or high-level semantic maps.

Low-level semantic maps represent object information and relations, often using scene graphs. Much of prior work focuses on learning-based methods using preexisting scene graphs [20]–[22]. Works that build a scene graph usually ignore the interactive robotic setting, instead restricting to single observations [23], [24] or predetermined observation sequences [25]–[27]. However, recently, Li *et al.* [28] built scene graphs with an exploration agent; we similarly build our map online during exploration. Other approaches to low-level semantic mapping produce metric maps capturing object information [29], [30]. Most similar to our method, a variety of approaches begin by performing low-level mapping (labeling objects), then back out high-level region categories [1]–[5]. In this work, we instead generate a region map online without explicitly representing low-level information.

High-level maps depict relations between regions or locations, which is less well-explored in the literature than low-level mapping. One key recent approach has been to generate hierarchical 3D scene graphs that can also capture room relations [31], [32]. Other approaches focus only on top-down semantic understanding for mobile robot navigation [33]. Most similar to our work, Sunderhauf *et al.* [34] used an external classifier to assign labels to RGB-D observations, then fused these into a semantic map. In contrast, our approach directly produces the semantic region map end-to-end, and can ground multiple labels spatially within a single observation. Thus, our work is also similar to the outdoor-focused work of Gan *et al.* [35], except that we leverage a vision-language model that enables more generalizable semantic mapping, instead of a pretrained semantic segmentation method with a fixed set of classes. We also note that such vision-language embeddings show promise for translating language commands into robot navigation trajectories [36], [37], but without creating a semantic map.

Scene Recognition. The task of correctly classifying a scene in an image has captured large interest in the Computer Vision community. Scene-centric datasets have been released to address this challenge. For example, Scene15 [38] was the first testbed for such a task but had a limited number of samples and classes. MIT Indoor67 [39] and SUN [40] enlarged the set of classes but the small number of samples limited the application of deep learning. More

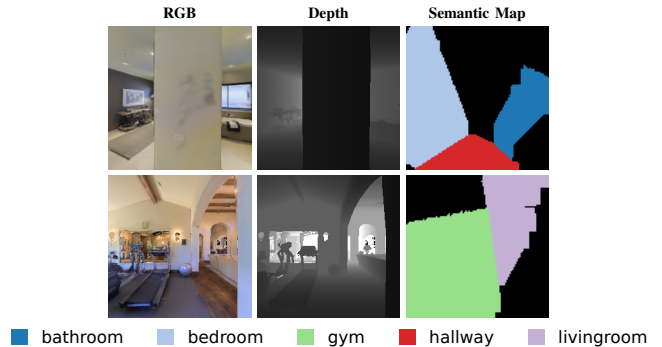


Fig. 2. Sample observations and corresponding top-down semantic maps from the extracted dataset. The upper row shows a common challenge faced in the dataset: occlusions and multiple semantic categories in a single image. Furthermore, on the right side of the upper image, the region is not obviously recognizable as a bathroom.

recently, Zhou *et al.* released Places [6], containing ≈ 8 million images. However scene-centric datasets are ill-suited for robot navigation, wherein observations are likely to contain occlusion or be uninformative (*e.g.* facing a wall). To address these shortcomings, we collect an offline dataset for the classification of rooms inside environments used for embodied navigation.

III. PROBLEM STATEMENT

The goal of this work is to produce top-down semantic maps depicting the different regions of an unseen environment while exploring it online. This is a challenging task because semantically mapping an unseen environment requires one to model and extract semantic information from the available observations while generalizing over the possible different appearances of each region. The goal is to create semantic maps that can be exploited for downstream navigation tasks. Overall, ISRM consists of two subtasks: *Region Classification* and *Region Mapping*.

Notation and Setup. In the following, we denote a probability distribution over x given y as $P(x | y)$. A set of n regions is denoted as $\mathcal{R} = \{r_1, \dots, r_n\}$, where the i^{th} region category is defined as $r_i \in \mathcal{R}$ (*e.g.*, $r_1 = \text{bathroom}$). Similarly, an RGB-D observation at timestep t extracted from the point of view of the agent is denoted as $s_t = \{s_t^{\text{rgb}}, s_t^{\text{d}}\}$. The pose (2-D position and heading) of the robot at time t is p_t , and the action taken at that time is a_t .

Region Classification. Suppose the robot has acquired an RGB observation s_t^{rgb} of the environment. In region classification, we seek to model a categorical distribution over regions given an RGB observation, *i.e.* $P(r_i | s_t^{\text{rgb}})$. Note that this formulation allows for ambiguity in classification, which frequently occurs in the dataset as shown in Figure 2.

Region Mapping. The output of region classification is not directly amenable to downstream tasks; hence, we seek to perform region mapping, to create a global map of semantic regions. In particular, suppose we are given an arbitrary 2D location x , and a history of observations up to time t , denoted $S = (s_0, \dots, s_t)$. We seek to model a distribution over possible regions, $P(r_i | x, S)$, and a distribution over static obstacle occupancy, $P(o | x, S)$, at the location x .

IV. PROPOSED METHOD

In this section, we first introduce the collection procedure and the features of our dataset of observation-map pairs. Then, we present our method for semantic region mapping and navigation.

A. Dataset

In order to train a model for region recognition inside photo-realistic environments we extract an off-line dataset of RGB-D observations from the point of view of the agent coupled with the corresponding top-down ground-truth region maps. Collecting a dataset rather than getting top-down maps directly from the simulator also allows us to efficiently shuffle the training samples and avoid an episode-based online setting, which would increase computational costs and lower the generalization capabilities of the trained method due to forgetting [41], [42].

1) *Extraction Procedure*: The offline dataset is extracted by running a state-of-the-art exploration method [9] using Habitat simulator [43] on the environments contained in Matterport3D dataset for a total of 236K samples. The observation-map pairs are saved taking into consideration the pose of the agent and are stored only if the agent has not visited the same position (within 0.1 meters) and orientation (within 0.1 radians \approx 6 degrees). We generate exploration episodes using the start position of the episode contained in the object-goal navigation task on Matterport3D dataset [44], which are collected on 61 and 11 unique building respectively for training and validation splits.

2) *Dataset Characteristics*: The resulting dataset is divided into 228K and 8K observation-map pairs for training and validation splits, respectively, which is a consequence of the original environment-level division of Matterport3D dataset. Further, we extract a training-val split consisting of a randomly sampled 5% of the items in the training set. The RGB-D observation resolution is 224×224 , with the depth observation that goes from 0 to 10 meters. The extracted map is $101 \times 101 \times C$ where C is the number of semantic region categories. We extract the maps maintaining the semantic labels used in the annotation of Matterport3D dataset [45]. Two samples from the dataset are shown in Figure 2.

B. Region Classification Module

To tackle the task of correctly classifying the current RGB observation s_t^{rgb} to a region label $r_i \in \mathcal{R}$, we employ the CLIP image-text model [7], which has been trained contrastively on large-scale image-text pairs and has shown good classification performance also in zero-shot scenarios. Nevertheless, CLIP still needs to be adapted to the task of region classification and to the distribution of indoor images to work properly, so we also design a finetuning strategy.

Overall, the architecture of the region classification module is composed of an image encoder that processes the RGB observation s_t^{rgb} to obtain the visual features ϕ_t^{rgb} , and a text encoder that takes as input the set of predefined region labels \mathcal{R} and extracts corresponding text features $\Phi^{\text{txt}} = [\phi_0^{\text{txt}}, \dots, \phi_R^{\text{txt}}]$. The predicted region label is derived

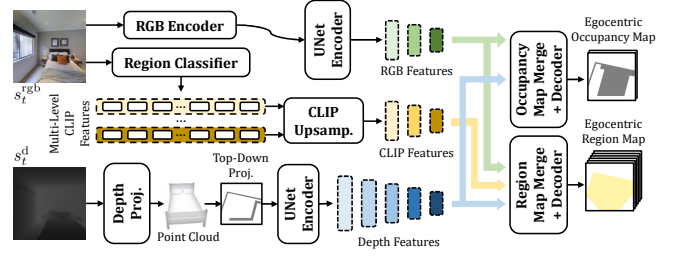


Fig. 3. Architecture of the proposed Semantic Region Mapper.

using the cosine similarity between the observation features ϕ_t^{rgb} and the text features of each region label Φ^{txt} :

$$r_i \quad \text{s.t.} \quad i = \underset{j}{\operatorname{argmax}} \frac{\phi_t^{\text{rgb}} \cdot \phi_j^{\text{txt}}}{\|\phi_t^{\text{rgb}}\| \cdot \|\phi_j^{\text{txt}}\|}. \quad (1)$$

The cosine similarity is used to model $P(r_i | x, S)$ as a categorical distribution.

1) *Multi-Modal Supervised Contrastive Loss*: To lower the computational requirements, we start with pretrained CLIP visual and text encoders and only finetune the projection toward the final embedding space. Differently from the image-text matching task on which CLIP has been trained, however, in region classification multiple samples in a mini-batch are likely to belong to the same textual label. The standard InfoNCE loss is incapable of handling such cases, which can result in reduced performance [8].

To overcome this issue and handle the case of having multiple samples with the same label, we devise a multi-modal adaptation of the Supervised Contrastive Loss [8]. Specifically, given a batch of N image features $\mathcal{V} = [\phi_0^{\text{rgb}}, \phi_1^{\text{rgb}}, \dots, \phi_{N-1}^{\text{rgb}}]$, we associate to each image the corresponding ground-truth region by using the label associated with the most pixels in the ground-truth top-down semantic map m^* . The resulting batch of N labels is used to extract the text features $\mathcal{T} = [\phi_0^{\text{txt}}, \phi_1^{\text{txt}}, \dots, \phi_{N-1}^{\text{txt}}]$, that are subsequently filtered to remove repeated entries, obtaining $\mathcal{T}' = [(\phi_0^{\text{txt}})', (\phi_1^{\text{txt}})', \dots, (\phi_{K-1}^{\text{txt}})']$. Visual and textual features, and their associated labels, are concatenated to obtain the features \mathcal{X} and labels \mathcal{Y} given as

$$\begin{aligned} \mathcal{X} &= [\phi_0^{\text{rgb}}, \dots, \phi_{N-1}^{\text{rgb}}, (\phi_0^{\text{txt}})', \dots, (\phi_{K-1}^{\text{txt}})'] \\ &= [\phi_0, \dots, \phi_{N-1}, \phi_N, \dots, \phi_{N+K-1}] \end{aligned} \quad (2)$$

$$\begin{aligned} \mathcal{Y} &= [y_0, \dots, y_{N-1}, y'_0, \dots, y'_{K-1}] \\ &= [y_0, \dots, y_{N-1}, y_N, \dots, y_{N+K-1}]. \end{aligned} \quad (3)$$

These are then used in the multi-modal supervised contrastive loss \mathcal{L}^{MSC} :

$$\mathcal{L}^{\text{MSC}} = \sum_{i=0}^{N+K-1} -\log \left(\frac{1}{|B(i)|} \sum_{b \in B(i)} \frac{\exp(\phi_i \cdot \phi_b / \tau)}{\sum_{a \in A(i)} \exp(\phi_i \cdot \phi_a / \tau)} \right), \quad (4)$$

where $\tau \in \mathbb{R}^+$ is a scalar temperature parameter, $A(i) \equiv \{0, \dots, N+K-1\} \setminus \{i\}$, $B(i) \equiv \{b \in A(i) : y_b = y_i\}$ is the set of indices of all positives in the batch distinct from i , and $|B(i)|$ is its cardinality.

C. Proposed Architecture

Once we have finetuned our region classification module, we can employ it within a learning-based region mapper paired with a hierarchical navigation policy. Our method first converts the RGB and depth observations into egocentric occupancy and region maps (*i.e.* distributions over occupancy and regions within the field of view of the robot). Then, we fuse the egocentric maps to produce a global map. Fig. 3 shows the architecture of the neural mapper and its modules.

1) *Egocentric Region Mapping*: The region mapper module builds a region-level semantic map of the environment while generating an obstacle occupancy grid. At each timestep, the RGB-D observation $s_t = (s_t^{\text{rgb}}, s_t^{\text{d}})$ is processed to extract a $L \times L \times (2+C)$ egocentric map m_t where the first two channels indicate the occupancy and exploration state of the currently observed region, and the last C channels are dedicated to the registration of observed region-level semantic classes. Each pixel of the map describes the state of a 5×5 cm area.

The RGB observation s_t^{rgb} is encoded using a ResNet-18 [46] encoder followed by a UNet [47] encoder. The depth observation s_t^{d} , along with intrinsic camera parameters, is used to generate a point cloud of the observed scene where each depth image pixel is projected to its 3D position with respect to the camera. The point cloud is then collapsed to obtain a top-down view s_t^{top} of the area observed by the agent. We use another UNet encoder to process the top-down view observation and extract the depth features.

2) *Semantic Feature Injection*: We employ the region classification model to produce semantically meaningful visual features of the current observation. Specifically, we extract features at different levels of the visual encoder of the region classifier, together with a CNN with transposed convolutional layers to upsample and align the spatial shape of the features to that of the output of the UNet encoders.

At this point, two parallel pipelines follow the generation of the occupancy map and the region map. We use two different CNNs to merge the multi-modal features obtained from the RGB and depth encoders, and the CLIP features. The merge module that generates the occupancy map uses RGB and depth features, while the other merge module also includes the CLIP features. The output of the merge modules is fed to two UNet decoders to produce the final egocentric occupancy map and region map.

Note that the occupancy state that is contained in the first two channels of the map m_t is produced following the method proposed in [19]. Therefore, our mapper is not limited to predicting the occupancy map of the visible space but tries to infer also occluded regions of the local map. However, the prediction of the current region map is limited to the visible area, to avoid projecting incorrect regions behind the walls and the obstacles of the visible region. In order to do so, we use the top-down projection extracted from the depth observation to mask the region map.

3) *Global Map via Egocentric Map Fusion*: During navigation, the global level map of the environment M_t with dimensionality of $G \times G \times (2 + C)$, where $G > L$, is

built using local maps m_t in an incremental fashion. At each timestep, the pose of the agent p_t is used to apply a corresponding rotation and translation to the local map. Finally, the transformed local map is fused with the global map M_t using a moving average (we compare against a Bayesian update in Sec. V).

4) *Navigation Module*: Following the literature on embodied exploration in unseen environments, we employ a hierarchical navigation policy that is responsible for predicting the atomic actions of the agent [9]. The policy is composed of a global policy, a deterministic planner, and a local policy. The global policy takes as input the current state of the global map M_t to generate long-term goals on the map g_t . The deterministic planner is in charge of generating the shortest trajectory using the observed occupancy between the agent and the global goal. A shorter-term goal l_t is sampled on the computed trajectory within 0.25m from agent’s position. The local policy is trained to reduce the distance from the local goal l_t while avoiding possible obstacles. Following the hierarchical design, the global goal g_t is updated every η timesteps, while the local goal l_t is updated if one of the following conditions holds: the global goal has been updated, the local goal is found to be in an occupied region, or the local goal has been reached.

D. Implementation Details

1) *Proposed Architecture*: We tested our model on the environments of Matterport3D using Habitat [43] simulator with an action space composed of three atomic actions: go ahead 0.25m, turn left 10° , and turn right 10° . The RGB-D observations used the region classification module and by the overall architecture have a 224×224 resolution, the minimum input size for a CLIP model. The mapper module extracts local maps with size $L = 101$, while the global maps have size $G = 2001$.

2) *Dataset Region Label Filtering*: Matterport3D contains a large variety of regions, some of which occur infrequently in the data (*e.g.*, bedroom is common, whereas spa is not). Furthermore, there are overlapping region types in the simulator (*e.g.*, porch/terrace/deck). To simplify our training procedure and avoid region ambiguities due to overlapping region labels, we postprocess our dataset to only contain the 14 most common labels, including other room. We also collapse overlapping region types into a single label. This results in the following set of filtered labels: $\mathcal{R} = \{ \text{bathroom, bedroom, closet, dining room, garage, gym, hallway, kitchen, library, living room, office, other room, outdoor, stairs} \}$.

V. EXPERIMENTS

In this section, we present experiments used to validate the proposed method. First, we assess the finetuned CLIP model used to condition the region mapper. Second, we study offline training of the mapper module on our collected dataset. Finally, we test a state-of-the-art exploration method that is equipped with our pretrained neural region mapper module. Note we are preparing our code to be posted as open-source online.

TABLE I
ACCURACY OF THE DIFFERENT CLIP MODELS ON ROOM CLASSIFICATION ON THE OFFLINE DATASET COLLECTED IN THE ENVIRONMENTS OF MATTERPORT3D.

Model	Off. MP3D	
	Train Val	Val
ResNet50 pretrained	29.48	26.33
ResNet50 + InfoNCE (only v. proj.)	72.22	32.18
ResNet50 + MSCL (only v. proj.)	86.85	33.30
ResNet50 + InfoNCE	85.66	30.34
ResNet50 + MSCL	85.87	31.02
ViT-B/32 pretrained	32.07	37.53
ViT-B/32 + InfoNCE (only v. proj.)	51.16	39.63
ViT-B/32 + MSCL (only v. proj.)	60.60	41.03
ViT-B/32 + InfoNCE	52.66	43.14
ViT-B/32 + MSCL	56.59	44.75
Human	-	49.58 ± 1.34

A. Region Classification finetuning

1) *Experimental Setup*: We perform an offline finetuning of the room classification module on the collected dataset, and evaluate on two variants of CLIP, namely the ones with a ResNet50 and a ViT-B/32 visual encoder. As these are two of the most lightweight CLIP variants, they are also appropriate for an embodied setting. We evaluate in terms of accuracy on the training-val and on the validation split of the collected offline dataset. The performance on the training-val split, which is a fraction of the training set, is reported to control overfitting.

2) *Baselines*: To assess the design of the finetuning methodology, we firstly compare to a baseline using an InfoNCE loss, instead of the proposed multi-modal supervised contrastive loss. Further, we also experiment when finetuning the final linear projections of both the visual and textual encoder, and when finetuning only the visual projection. In parallel, to give an idea of the difficulty of the task, we tested human performance in a user study dividing the validation unseen of the offline dataset into 16 parts and asking 16 people unaware of the environments in Matterport3D to classify each observation with a region label.

3) *Results and Discussion*: As shown in Table I, pre-trained CLIP models show a low zero-shot classification performance, with a validation accuracy ranging from 26% (in the case of ResNet50) to 37.5% (in the case of the ViT-based backbone). This highlights that the specificity of the task requires better alignment of the visual appearances associated with each class, and motivates the need for finetuning. We also note that trying to adapt such a model online with the navigation of the agent could result in poor performance due to the high correlation between observations in an episode. Therefore, shuffling the training pairs used for finetuning is beneficial for the generalization capabilities of the model.

Using the multi-modal supervised contrastive loss (MSCL) provides a considerable performance gain, while finetuning the textual encoder only provides better results on one architecture, *i.e.* ResNet50. We notice that, while finetuning provides an improvement over the pre-trained backbone, it also suffers overfitting, which might be worth investigating

TABLE II
REGION-LEVEL MAPPING PERFORMANCE ON THE OFFLINE MATTERPORT3D DATASET.

Model	Off. MP3D			
	Train Val		Val	
	Acc (↑)	IoU (↑)	Acc (↑)	IoU (↑)
ResNet50 pretrained + projection	2.22	0.19	2.42	0.62
ViT-B/32 pretrained + projection	3.26	0.82	3.96	0.62
ResNet50 fin. (o.v.p.) + projection	14.86	7.28	2.81	0.62
ViT-B/32 fin. + projection	16.81	1.64	5.29	0.66
ResNet50 fin. w. repeated features	47.08	16.84	31.41	16.45
ResNet50 fin. w. spatial features	46.68	16.65	31.31	15.10
ViT-B/32 fin. w. repeated features	44.57	16.71	31.10	17.13
ViT-B/32 fin. w. spat. feat. (Ours)	54.24	28.56	31.19	18.40

in future architectural and training variants. Overall, finetuning the visual projection provides a validation accuracy of 33.3% for ResNet50 and 44.75% for ViT-B/32, highlighting the benefit of finetuning the image-text model for region classification.

B. Region-Level Mapping Training

1) *Experimental Setup*: We then move to the experiments using the region-level mapper evaluating it in terms of pixel-level mapping accuracy and mean Intersection-over-Union with respect to ground-truth region maps extracted from the offline dataset. Also, in this case, the evaluation is conducted on both the train-validation and validation splits.

2) *Baselines*: We consider three baselines. First, we consider a simple baseline in which the most probable region class, as predicted by the image-text model, is directly employed to build the egocentric map, without employing the region-map merge component. In this case, the predicted region class is projected using the depth map directly on the egocentric map. Furthermore, we report the results obtained when employing the CLIP module finetuned with MSCL as part of our semantic region mapper. When injecting the features extracted from the CLIP visual encoder, we devise two different alternatives as baselines: our second baseline repeats the final feature vector spatially, to match the spatial shape of the UNet encoders, and our third baseline employs the CNN with transposed convolutions to upsample and match spatial resolutions.

3) *Results and Discussion*: Results are reported in Table II. First, we observe that repeating the most probable semantic class, as predicted from the egocentric observation of the agent, provides low performance. For instance, a pre-trained CLIP with ViT-B/32 reaches around 4% accuracy on the validation set. Applying our finetuning strategy increases performance, up to 5% accuracy on the validation set. Injecting CLIP features in the semantic region mapper instead provides significantly better results, highlighting that a proper strategy to translate CLIP predictions into an egocentric map is needed. In this regard, we notice that the spatial features extracted from a ResNet50 are less powerful than those extracted from the ViT-based backbone. When using the ResNet50 backbone, we notice that simply repeating the global feature vector is better than employing intermediate

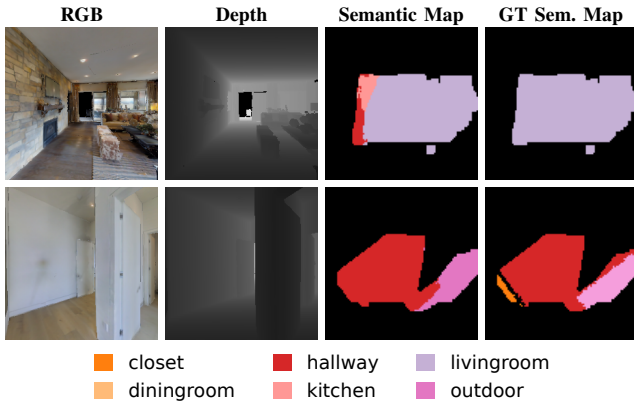


Fig. 4. Samples of generated maps with corresponding ground truth and visual inputs on the environments of Matterport3D. Note that the first row shows that the model defines the region between the TV and the couches as a hallway (in red).

TABLE III
REGION-LEVEL MAPPING PERFORMANCE ON THE OFFLINE MATTERPORT3D DATASET USING DIFFERENT INPUT MODALITIES.

Model	Off. MP3D			
	Train Val		Val	
	Acc (↑)	IoU (↑)	Acc (↑)	IoU (↑)
Ours RGB-only	27.42	15.48	15.75	6.52
Ours Depth-only	46.34	22.83	28.88	17.32
Ours	54.24	28.56	31.19	18.40

and spatial-aware features. On the contrary, the ViT-based backbone provides better spatial features. Overall, the proposed models achieve up to 31.4% accuracy and up to 18.4% IoU on the validation set.

4) *Qualitative Samples*: Following the quantitative analysis of the map generation of the region mapper, in Figure 4 we present some samples of the output of the region map decoder with the corresponding input acquired on the environments of Matterport3D (MP3D) dataset [45].

5) *Ablation Study*: To validate the contribution of using both RGB and depth modalities for the input of the neural mapper, in Table III we compare our approach using ViT-B/32 backbone and spatial features with two baselines using each modality singularly. The model using RGB-D surpasses the other baselines, indicating that using both RGB and depth information is beneficial for the final results.

C. Online Mapping Experiments

1) *Experimental Setup*: The online mapping experiment is performed using the Habitat simulator [43] with Matterport3D (MP3D) [45] environments. Among the available datasets for photo-realistic embodied navigation like Gibson [48], Habitat-Matterport3D (HM3D) [49], only Matterport3D contains room annotations. We test our mapping approach on board of a state-of-the-art exploration method [9] to evaluate the agent’s ability to correctly classify the observed regions. We compare against a baseline using a perfect object detector that maps the environment by propagating the region labels associated with the closest object characterizing a specific region (bed \rightarrow bedroom) similar to [1]–[5]. We

TABLE IV
ONLINE MAPPING PERFORMANCE OVER THE MATTERPORT3D DATASET.

Model	MP3D Val		
	maskAcc (↑)	ovrAcc (↑)	IoU (↑)
Baseline w. oracle object detection (c.f. [1]–[5])	26.58	8.63	19.44
Baseline w. pretrained scene detector (c.f. [34])	28.46	9.42	21.03
ViT-B/32 (repeated features)	33.67	18.60	20.10
ViT-B/32 (spatial features) and Bayesian update	13.08	10.58	6.48
ViT-B/32 (spatial features) and noise (ours)	32.73	13.74	22.08
ViT-B/32 (spatial features) (ours)	40.81	22.70	24.68

also test a baseline based on Sunderhauf *et al.* [34], but using a ResNet50 model trained on the more recent Places365 dataset [6].

2) *Global Room-Level Mapping*: In Table IV we evaluate the performance of our mapper in the online setting. For our approach, we use CLIP with ViT-B/32 backbone, since it outperformed ResNet in the offline setting. We test both repeated and spatial features. We also compare a Bayesian map update as opposed to a moving average. Finally, to test the validity of assuming perfect pose and depth information in our formulation, we also compare with a noisy baseline using Habitat’s built-in noise models for the pose sensor [43] and the depth camera [50]

We report the results in terms of IoU with respect to the entire ground-truth semantic map, the pixel-level accuracy restricted to the area explored by the agent (maskAcc), and the overall pixel-level accuracy over the entire navigable area (ovrAcc). As expected, the results are generally in line with the offline setting, with the ViT-B/32 backbone with spatial features providing the best results (masked accuracy of 40.81% and IoU of 24.68%). The baseline using oracle object detection achieves only 26.6% accuracy, since areas close to characteristic objects often belong to different rooms. We also find that a moving average is more robust than a Bayesian update against spurious and rapidly-changing high-confidence classifications from timestep to timestep (e.g., outputting `bedroom` with high confidence at time t , then `bathroom` at $t + 1$), which is a well-known problem with neural network classifiers [51]. Finally, we see that noise in pose and depth does produce some impact on performance, but our method with noise still outperforms the other noise-free baselines.

VI. CONCLUSION

We proposed a method for semantic region mapping for Embodied navigation in indoor environments. Our research is motivated by the need to endow agents with a semantic mapping of their surroundings, which is currently under-explored and essential to provide future embodied agents with semantic understanding capabilities. Our approach comprises a region classification module, based on a finetuned CLIP model, a region mapping architecture with semantic feature injection, and a navigation module, aware of high-level region semantics. Experiments, conducted on a novel dataset for the region mapping task, and on the Matterport 3D dataset, show that endowing a robot with high-level semantic understanding can convey an advantage over typical low-level object-based mapping.

REFERENCES

- [1] A. C. Hernández, C. Gómez, R. Barber, and O. M. Mozos, “Object-based probabilistic place recognition for indoor human environments,” in *Proceedings of the IEEE International Conference on Control, Artificial Intelligence, Robotics & Optimization*, 2018.
- [2] M. Brucker, M. Durner, R. Ambruş, *et al.*, “Semantic labeling of indoor environments from 3d rgb maps,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2020.
- [3] D. Fernandez-Chaves, J.-R. Ruiz-Sarmiento, N. Petkov, and J. Gonzalez-Jimenez, “From object detection to room categorization in robotics,” in *Proceedings of the International Conference on Applications of Intelligent Systems*, 2020.
- [4] X. Qi, W. Wang, M. Yuan, *et al.*, “Building semantic grid maps for domestic robot navigation,” *International Journal of Advanced Robotic Systems*, 2020.
- [5] P. Wang, J. Cheng, and W. Feng, “An approach for construct semantic map with scene classification and object semantic segmentation,” in *Proceedings of the IEEE International Conference on Real-time Computing and Robotics*, 2018.
- [6] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, “Places: A 10 million image database for scene recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.
- [7] A. Radford, J. W. Kim, C. Hallacy, *et al.*, “Learning transferable visual models from natural language supervision,” in *Proceedings of the International Conference on Machine Learning*, 2021.
- [8] P. Khosla, P. Teterwak, C. Wang, *et al.*, “Supervised contrastive learning,” in *Advances in Neural Information Processing Systems*, 2020.
- [9] R. Bigazzi, F. Landi, S. Cascianelli, L. Baraldi, M. Cornia, and R. Cucchiara, “Focus on impact: Indoor exploration with intrinsic motivation,” *IEEE Robotics and Automation Letters*, 2022.
- [10] R. C. Smith and P. Cheeseman, “On the Representation and Estimation of Spatial Uncertainty,” *The International Journal of Robotics Research*, 1986.
- [11] S. Thrun, “Probabilistic Robotics,” *Communications of the ACM*, 2002.
- [12] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.
- [13] J. Zhang, L. Tai, M. Liu, J. Boedecker, and W. Burgard, “Neural slam: Learning to explore with external memory,” *arXiv preprint arXiv:1706.09520*, 2017.
- [14] D. S. Chaplot, D. Gandhi, S. Gupta, A. Gupta, and R. Salakhutdinov, “Learning to explore using active neural slam,” in *Proceedings of the International Conference on Learning Representations*, 2019.
- [15] N. Savinov, A. Dosovitskiy, and V. Koltun, “Semi-parametric topological memory for navigation,” in *Proceedings of the International Conference on Learning Representations*, 2018.
- [16] N. Savinov, A. Raichuk, R. Marinier, *et al.*, “Episodic curiosity through reachability,” in *Proceedings of the International Conference on Learning Representations*, 2019.
- [17] E. Parisotto and R. Salakhutdinov, “Neural map: Structured memory for deep reinforcement learning,” in *Proceedings of the International Conference on Learning Representations*, 2018.
- [18] J. F. Henriques and A. Vedaldi, “Mapnet: An allocentric spatial memory for mapping environments,” in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8476–8484.
- [19] S. K. Ramakrishnan, Z. Al-Halah, and K. Grauman, “Occupancy Anticipation for Efficient Exploration and Navigation,” in *Proceedings of the European Conference on Computer Vision*, 2020.
- [20] R. Druon, Y. Yoshiyasu, A. Kanezaki, and A. Watt, “Visual object search by learning spatial context,” *IEEE Robotics and Automation Letters*, 2020.
- [21] Z. Zeng, A. Röfer, and O. C. Jenkins, “Semantic linking maps for active visual object search,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2020.
- [22] H. Du, X. Yu, and L. Zheng, “Learning object relation graph and tentative policy for visual navigation,” in *Proceedings of the European Conference on Computer Vision*, 2020.
- [23] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh, “Graph r-cnn for scene graph generation,” in *Proceedings of the European Conference on Computer Vision*, 2018.
- [24] J. Gu, H. Zhao, Z. Lin, S. Li, J. Cai, and M. Ling, “Scene graph generation with external knowledge and image reconstruction,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.
- [25] I. Armeni, Z.-Y. He, J. Gwak, *et al.*, “3d scene graph: A structure for unified semantics, 3d space, and camera,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.
- [26] J. Wald, H. Dharmo, N. Navab, and F. Tombari, “Learning 3d semantic scene graphs from 3d indoor reconstructions,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.
- [27] S.-C. Wu, J. Wald, K. Tateno, N. Navab, and F. Tombari, “Scenegrphfusion: Incremental 3d scene graph prediction from rgb-d sequences,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.
- [28] X. Li, D. Guo, H. Liu, and F. Sun, “Embodied semantic scene graph generation,” in *Proceedings of the Conference on Robot Learning*, 2022.
- [29] D. S. Chaplot, D. P. Gandhi, A. Gupta, and R. R. Salakhutdinov, “Object goal navigation using goal-

- oriented semantic exploration,” in *Advances in Neural Information Processing Systems*, 2020.
- [30] V. Cartillier, Z. Ren, N. Jain, S. Lee, I. Essa, and D. Batra, “Semantic mapnet: Building allocentric semantic maps and representations from egocentric views,” *Proceedings of the Conference on Artificial Intelligence*, 2020.
- [31] A. Rosinol, A. Violette, M. Abate, *et al.*, “Kimera: From slam to spatial perception with 3d dynamic scene graphs,” *The International Journal of Robotics Research*, 2021.
- [32] N. Hughes, Y. Chang, and L. Carlone, “Hydra: A real-time spatial perception system for 3d scene graph construction and optimization,” in *Robotics: Science and Systems*, 2022.
- [33] M. Narasimhan, E. Wijmans, X. Chen, *et al.*, “Seeing the un-scene: Learning amodal semantic maps for room navigation,” in *Proceedings of the European Conference on Computer Vision*, 2020.
- [34] N. Sünderhauf, F. Dayoub, S. McMahan, *et al.*, “Place categorization and semantic mapping on a mobile robot,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2016.
- [35] L. Gan, R. Zhang, J. W. Grizzle, R. M. Eustice, and M. Ghaffari, “Bayesian spatial kernel smoothing for scalable dense semantic mapping,” *IEEE Robotics and Automation Letters*, 2020.
- [36] N. M. M. Shafiullah, C. Paxton, L. Pinto, S. Chintala, and A. Szlam, “Clip-fields: Weakly supervised semantic fields for robotic memory,” in *Workshop on Language and Robotics at CoRL*, 2022.
- [37] D. Shah, B. Osiński, brian ichter, and S. Levine, “Lmnav: Robotic navigation with large pre-trained models of language, vision, and action,” in *Proceedings of the Conference on Robot Learning*, 2022.
- [38] A. Oliva and A. Torralba, “Modeling the shape of the scene: A holistic representation of the spatial envelope,” *International Journal of Computer Vision*, 2001.
- [39] A. Quattoni and A. Torralba, “Recognizing indoor scenes,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2009.
- [40] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, “Sun database: Large-scale scene recognition from abbey to zoo,” in *2010 IEEE computer society conference on computer vision and pattern recognition*, IEEE, 2010, pp. 3485–3492.
- [41] M. McCloskey and N. J. Cohen, “Catastrophic interference in connectionist networks: The sequential learning problem,” in *Psychology of Learning and Motivation*, 1989.
- [42] A. Robins, “Catastrophic forgetting, rehearsal and pseudorehearsal,” *Connection Science*, 1995.
- [43] M. Savva, A. Kadian, O. Maksymets, *et al.*, “Habitat: A platform for embodied ai research,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.
- [44] D. Batra, A. Gokaslan, A. Kembhavi, *et al.*, “Objectnav revisited: On evaluation of embodied agents navigating to objects,” *arXiv preprint arXiv:2006.13171*, 2020.
- [45] A. Chang, A. Dai, T. Funkhouser, *et al.*, “Matterport3d: Learning from rgb-d data in indoor environments,” in *Proceedings of the International Conference on 3D Vision*, 2017.
- [46] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.
- [47] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention*, 2015.
- [48] F. Xia, A. R. Zamir, Z. He, A. Sax, J. Malik, and S. Savarese, “Gibson env: Real-world perception for embodied agents,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.
- [49] S. K. Ramakrishnan, A. Gokaslan, E. Wijmans, *et al.*, “Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai,” in *Advances in Neural Information Processing Systems*, 2021.
- [50] S. Choi, Q.-Y. Zhou, and V. Koltun, “Robust reconstruction of indoor scenes,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015.
- [51] A. Nguyen, J. Yosinski, and J. Clune, “Deep neural networks are easily fooled: High confidence predictions for unrecognizable images,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 427–436.