Learning Navigational Visual Representations with Semantic Map Supervision

Yicong Hong\textsuperscript{1,\textdagger} Yang Zhou\textsuperscript{1} Ruiyi Zhang\textsuperscript{1} Franck Dernoncourt\textsuperscript{1} Trung Bui\textsuperscript{1} Stephen Gould\textsuperscript{2} Hao Tan\textsuperscript{1}
\textsuperscript{1}Adobe Research \textsuperscript{2}The Australian National University

\texttt{mr.yiconghong@gmail.com}

stephen.gould@anu.edu.au, \{yazhou, ruizhang, dernonco, bui, hatan\}@adobe.com

Project URL: \url{https://github.com/YicongHong/Ego2Map-NaViT}

Abstract

Being able to perceive the semantics and the spatial structure of the environment is essential for visual navigation of a household robot. However, most existing works only employ visual backbones pre-trained either with independent images for classification or with self-supervised learning methods to adapt to the indoor navigation domain, neglecting the spatial relationships that are essential to the learning of navigation. Inspired by the behavior that humans naturally build semantically and spatially meaningful cognitive maps in their brains during navigation, in this paper, we propose a novel navigational-specific visual representation learning method by contrasting the agent’s egocentric views and semantic maps (Ego\textsuperscript{2}-Map). We apply the visual transformer as the backbone encoder and train the model with data collected from the large-scale Habitat-Matterport3D environments. Ego\textsuperscript{2}-Map learning transfers the compact and rich information from a map, such as objects, structure and transition, to the agent’s egocentric representations for navigation. Experiments show that agents using our learned representations on object-goal navigation outperform recent visual pre-training methods. Moreover, our representations significantly improve vision-and-language navigation in continuous environments for both high-level and low-level action spaces, achieving new state-of-the-art results of 47\% SR and 41\% SPL on the test server.

1. Introduction

Visual representations for navigation should capture the rich semantics and the complex spatial relationships of the observations, which helps the agent to recognize visual entities and its transition in space for effective exploration. However, previous works usually adopt visual backbones which only focus on capturing the semantics of a static image, ignoring its connection to the agent or the correspondence to other views in a continuous environment [3, 9, 40, 42, 51, 68, 98]. One common approach is to apply encoders pre-trained for object/scene classification (e.g. ResNet [39] on ImageNet [80]/Places365 [102]), object detection (e.g. RCNN [38, 78] on VisualGenome [52]/MSCOCO [57]) or semantic segmentation (e.g. RedNet [45] on SUN RGBD [87]), or more recently, to use CLIP [72], which is trained for aligning millions of images and texts to encode agent’s RGB observations [31, 46, 84]. Despite the increasing generalization ability of the features and rising zero-shot performance on novel targets, there still exists a large visual domain gap between these features and the features suitable for navigation since they lack the expressiveness of spatial relationships. For example, the connection between time-dependent observations and the correspondence between egocentric views and the spatial structure of an environment, which are important to decision making during navigation.

We suggest that there are two main difficulties in learning navigational-specific visual representations in previous research: First, there lacked a large-scale and realistic dataset of indoor environments. Popular datasets such as Matterport3D [8] and Gibson [95] provide traversable scenes rendered from real photos, but either the number of scenes is very limited, or the quality of the 3D scan is low. Synthetic datasets such as ProcTHOR [22] contain 10K generated scenes, but they are unrealistic and simple and do not capture the full complexity of real-world environments. Second, fine-tuning visual encoders while learning to navigate is very expensive because of the long traveling horizon, especially in tasks that require extensive exploration [4, 53, 63, 86]. Moreover, due to the scene scarcity, fine-tuning visual encoders is unlikely to generalize well to the novel environments. To address these problems, a large amount of research has been dedicated to augmenting the environments, either editing the agent’s observations...
requires an agent to navigate in photo-realistic environments (R2R-CE) task, which on the Room-to-Room Vision-and-Language Navigation in Continuous Environments (R2R-CE) [3, 51] task, which aims to produce robust and generalizable visual representations. We found that the proposed Ego²-Map features significantly boost the agent’s performance, obtaining +3.56% and +5.10% absolute SPL improvements over the CLIP baseline [72] under the settings of high-level and low-level action spaces, respectively, and achieves the new best results on the R2R-CE test server. Moreover, our experiments show that Ego²-Map learning also outperforms other visual representation learning methods such as OVRL [96] on the Object-Goal Navigation task (ObjNav), suggesting the strong generalization potential of the proposed methods.

2. Related Work

Visual Navigation A great variety of scenarios have been proposed to learn visual navigation in photo-realistic environments [8, 21, 54, 82, 86, 89, 95] with different modalities of inputs, targets and action spaces [3, 4, 11, 13, 51, 66, 70, 94]. Due to the distinct nature of the tasks, diverse methods are investigated accordingly. For instance, in Object-Nav, which only provides a high-level goal and requires exploration, mapping-based methods are frequently applied [10, 32, 60, 73, 105], whereas in vision-and-language navigation (VLN) [3], large vision-language models are employed to match instructions and agent’s observations [42, 61, 71, 83, 103] to perform panoramic actions [30]. Recently, there is an emerging trend of scaling up the training data to address the common data scarcity issue, either in terms of the number of environments [22, 74] or the amount of supervision [16, 76, 93]. Unlike most of the previous works, we focus on improving the visual backbone, aiming to produce robust and generalizable visual representations specialized for navigation.

Visual Representations for Embodied AI Recent years have witnessed a trend of moving away from visual encoders pre-trained for object classification in Embodied AI due to their inefficiency in representing complex real-world scenes or mapping to actions. Instead, large vision-language models such as CLIP which demonstrates strong zero-shot performance across visual domains is widely applied [20, 72], enhancing the semantic representations in robotic manipulation [85], assisting 3D trajectory modification and speed control [5], benefiting the language-conditioned view-selection problem in VLN [83, 84] as well as improving other control and navigation results [31, 46, 67]. Moreover, EmbCLIP shows that the CLIP features provide much better semantic and geometric primitives such as...
Figure 1: Network architecture. The Semantic MapNet [7] is applied to draw a semantic map using observations captured from a source to a target position. Then, two learnable visual encoders are applied to encode the egocentric views and the map, respectively, followed by non-linear headers $\Pi_v$ and $\Pi_m$ to learn the Ego$^2$-Map contrastive objective $L_c$. $f_s$ and $f_t$ are the paired view features of a trajectory (see §3.2). Dash arrows indicate pre-processing without gradient update.

object presence, object reachability, and free space that are valuable to Embodied AI [46]. In terms of self-supervised visual representation learning for indoor navigation, recent works that are the most relevant to ours include OVRL [96] which applies knowledge distillation based on DINO [6], EPC which predicts masked observations from a trajectory [75], and CRL which jointly learns a visual encoder with a policy network that maximizes the representation error [27]. In contrast to all previous works, this paper explores a novel idea of encoding the explicit structural and semantic information available in maps implicitly in the visual encoder thereby facilitating efficient and effective generalization across different visual navigation tasks and models.

3. Contrastive Learning between Egocentric Views and Semantic Maps (Ego$^2$-Map)

We will first describe the overall network architecture for training the visual encoders (§3.1), followed by the details of Ego$^2$-Map objectives (§3.2), as well as two other widely applied spatial-aware auxiliary tasks which we investigated with our Ego$^2$-Map learning (§3.3). Then, we will talk about the data collection (§3.4) and network training (§3.5) processes.

3.1. Network Architecture

We build the network for Ego$^2$-Map learning based on a ViT-B/16 model (default initialized from CLIP [72], whose effect will be studied in §4.3). Unlike previous methods for visual navigation, which mostly employ two independent encoders to process RGB and depth images, we investigate a compact representation by feeding RGB+Depth as four-channel inputs (Figure 1). To merge the two visual modalities, we use two separate convolutional layers to encode the RGB channels $I_{rgb}$ and the depth map $I_d$, respectively, followed by a token-wise concatenation and a non-linear projection $\Pi_v$ to merge the resulting feature maps before feeding to the transformer layers as

\[ v_{rgb} = \text{Conv}_{rgb}(I_{rgb}), \quad v_d = \text{Conv}_d(I_d) \] (1)

and

\[ f = \text{ViT}(\Pi_v [v_{rgb}; v_d]) \] (2)

Exploiting Visual Features

A great variety of proxy tasks and auxiliary objectives have been applied to exploit information from the visual data which is beneficial to navigation. For instance, Ye et al. [98, 99] promote understanding of spatiotemporal relations of observations by estimating the timestep difference between two views, Qi et al. [69] predict room-types from object features to improve scene understanding, and Se et al. [81] determine the furthest reachable waypoint on a topological graph using the node observations. To extract the geometric information, Gordon et al. [34] use the encoded visual features to predict depth and surface normals, reconstruct RGB input, and forecast the visual features by taking certain actions. Depth prediction has also been studied by Mirowski et al. [64], Desai et al. [24] and Chattopadhyay et al. [12], along with learning the inverse dynamics by predicting an action taken between two sequential observations [37, 64]. Complementary to the previous methods for interpreting visual features, our Ego$^2$-Map learning guides a visual encoder to produce representations with expressive spatial information.
The pooled features $f$ of the RGBD encoder are applied for learning all spatial-aware objectives $L_\theta$, $L_d$ and $L_c$, which will be specified in the following.

### 3.2. Ego$^2$-Map Contrastive Objective

The motivation of this task is to introduce the information within a map to single-view features, offering the agent high-level semantic and spatial clues to navigate towards distant targets. Specifically, we build positive samples by coupling a views-pair ($I_s$ and $I_t$ from the two endpoints of a path) with a top-down semantic map $M$ that represents the agent’s transition and observations along the path, while using mismatched views-pairs and maps of different routes or environments as the negatives. Each semantic map is generated by the off-the-shelf Semantic MapNet [7], using the RGBD observations collected by an agent traveling from the source view $I_s$ to the target view $I_t$ via the shortest path$^2$. The semantic map provides abundant information about the open space, obstacles, unexplored regions, observed objects, and the agent’s action, which we consider as highly valuable and compact representations for agent navigation (see maps in Figure 2). To encode the source view $I_s$ to the target view $I_t$ via the shortest path, we apply an additional ViT-B/32 [25] with the last three transformer layers unfrozen to adapt the map images. Then, the egocentric features of the two views $f_s$ and $f_t$, and the map features $f_m$ will be passed to two MLP (multi-layer perceptrons) headers $\Pi_i$ and $\Pi_m$, respectively, to compute an alignment score as:

$$ c^I = \Pi_i[f_s; f_t], \quad c^M = \Pi_m[f_m] $$

and

$$ \langle c^I, c^M \rangle = \frac{c^I \cdot c^M}{\|c^I\| \|c^M\|} $$

Then, the InfoNCE loss [65, 101] is applied for Ego$^2$-Map contrastive learning. For each $j$-th views-map pair in a minibatch of size $N$, we have

$$ L_{c,j} = L_{c,j}^{I \rightarrow M} + L_{c,j}^{M \rightarrow I} $$

where the views to maps ($I \rightarrow M$) loss can be expressed as

$$ L_{c,j}^{I \rightarrow M} = -\log \frac{\exp(\langle c_j^I, c_j^M \rangle / \tau)}{\sum_{k=1}^{N} \exp(\langle c_j^I, c_k^M \rangle / \tau)} $$

and the map to views loss $L_{c,j}^{M \rightarrow I}$ likewise. Parameter $\tau \in \mathbb{R}_+$ denotes a learnable temperature.

$^2$Each resulting map is unique because the observations are determined by the unique path (actions) for connecting two views. For example, in Figure 2, only the colored regions in the map ($M$) are the areas seen by the agent when traveling from the source to the target positions.

### 3.3. Other Spatial-Aware Objectives

We investigate two additional widely applied proxy tasks with our Ego$^2$-Map learning. Note that we do not claim any novelty for applying these tasks but focus on studying their effect on training the representations.

#### Angular Offset Prediction
Given two images taken from random headings at the same position, the task predicts the angular offset between them to facilitate the modeling of correspondence between views. A similar proxy task has been applied in previous works, which predicts discrete angles and uses an extra directional encoding to imply orientation [15] or regresses the agent’s turning angle conditioned on language during navigation [104]. Specifically, we pass the two pooled RGBD features $f_{\theta_0}$ and $f_{\theta_1}$, corresponding to images at two orientations of the same viewpoint, into a learnable non-linear header $\Pi_\theta$ to predict the offset as $\theta^p = \Pi_\theta[f_{\theta_0}; f_{\theta_1}]$. The task is learned by minimizing the mean squared error $L_\theta = E[(\theta^p - \theta^*)^2]$, where $\theta^*$ is the ground-truth angular difference between $[-\pi, \pi]$, denoting either clockwise or counter-clockwise rotation whichever is closer to encourage agent’s efficient rotation.

#### Explorable Distance Prediction
To benefit the searching of explorable regions and assist obstacle avoidance, the task estimates the agent’s maximal distance to forward without being blocked by any obstacle [11, 36]. Using the same notations as above, the module regresses a value from an RGBD image as $d^p = \Pi_d[f]$, and is trained with supervision $L_d = E[(d^p - d^*)^2]$. We cap the ground-truth distance $d^*$ in the range of 0.5 to 5.0 meters to match the common range of the agent’s depth sensors.

### 3.4. Data Collection

We follow the train-validation-test split of the HM3D scenes [74] and use the first 800 environments for learning. We randomly sample 252,537 viewpoint positions from the environments (see Figure 3) and render more than a million RGBD images from those positions to create 500,000 ($I_{\theta_0}, I_{\theta_1}$) views pairs as well as 500,000 ($I_s, I_t, M$) triplets for learning $L_\theta$ and $L_d$, respectively, while using all the images to learn $L_d$. The positions are sampled such that all points must be located in the open space and the minimal geodesic distance between any two points is greater than 0.40 meters. Each image collected for Ego$^2$-Map contrastive learning is either a unique source or target view in all trajectories, and each trajectory is created by computing the shortest path between two randomly paired views within a 7 meters range. We feed the RGBD images captured from a trajectory to the Semantic MapNet [7] for drawing the corresponding semantic map. Note that the Semantic MapNet is trained on the MP3D environments [8], but we found that
Figure 2: Illustration of Ego$^2$-Map contrastive learning. The method encodes the paired source and target egocentric views ($I_s, I_t$) into feature $c^I$, then learns to align it with the feature $c^M$ of the corresponding top-down map $M$. Negative view pairs ($I'_s, I'_t$) and maps ($M'$) are sampled from different trajectories or different environments, which are pushed away from the reference pair. The blue dashed line on the left part of the figure indicates the agent’s trajectory, which is visualized with directional color in map $M$. The map is also powered with semantic information (denoted at the bottom right).

Figure 3: Illustration of an HM3D environment scan and the sampled positions (blue points) in open space (gray area).

It also generalizes well to the HM3D scenes. We refer to the Appendix for more sampling and dataset creation details.

3.5. Training

Implementation Details We initialize the RGBD encoder and the map encoder of our visual representation network with CLIP, a visual transformer pre-trained for aligning image-language pairs [72]. Specifically, the additional depth convolutional layer and the projection layer in the RGBD encoder are initialized randomly. All network variants in our experiments are trained with batch size 994 for 100 epochs using the AdamW optimizer [59]. After all training iterations, only the RGBD encoder will be applied as the visual backbone for navigation agents. Image augmentations are applied to all views and maps.

Optimization The overall pre-training strategy minimizes the sum over all three losses, $\mathcal{L}_c$, $\mathcal{L}_\theta$ and $\mathcal{L}_d$. All losses are equally weighted and optimized simultaneously in each iteration. After all training steps, evaluate the model with batch size 128 on 10,000 novel ($I_s, I_t, M$) triplets shows 92.02% views to maps ($I \rightarrow M$) and 92.03% maps to view ($M \rightarrow I$) alignment accuracy, indicating the correspondence between two modalities has been learned. Please see Appendix for more pre-training statistics and results.

4. Downstream Experiments

We evaluate our Ego$^2$-Map representations mainly on the R2R-CE navigation [3, 51], as addressing the task highly relies on exploiting the semantic and structural information from observations and ground to language instructions. We also show results on ObjNav [4] to compare with other recent visual representation learning approaches.

4.1. R2R-CE Setup

Our experiments consider two major setups of action spaces, either applying panorama inputs and execute high-level decisions by selecting images pointing towards navigable directions ($A_{High}$), or using egocentric views and perform low-level controls ($A_{Low}$). For $A_{High}$, we adopt the candidate waypoint predictor proposed by Hong et al. [41], which generates navigable waypoint positions around the agent at each time step. Once a waypoint is selected, the agent will move to the position with low-level

---

3High-level VLN agents apply 36 egocentric images to represent its panoramic vision [3], hence, our Ego$^2$-Map features are applicable.
controls. For $A_{Low}$, an agent is only allowed to do a single low-level action at each step, including Turn Left/Right 15°, Forward 0.25 meters, and Stop. In all experiments, we apply the VLN*BERT agent [32], a vision-language transformer pre-trained for navigation, as the policy network. To use our Ego²-Map representations, we simply replace the agent’s original visual encoders (two ResNets trained on ImageNet [39] and trained in Point-Nav [94] for RGB and depth, respectively) with our RGBD encoder ($\theta$). Comparing the runtime efficiency, the original encoder has around 50M parameters and the speed is about 4 GFLOPs/image, whereas our method is larger (86M parameters) but faster (20 GFLOPs/image). All visual encoders in our experiments are frozen in navigation.

R2R in continuous environment (R2R-CE) is established over the Matterport3D environments [8] based on the Habitat simulator [32]. The dataset contains 61 scans for training; 11 and 18 scans for validation and testing, respectively. R2R-CE evaluates the agent’s performance using multiple metrics, including trajectory length (TL), navigation error (NE): the final distance between the agent and the target, normalized dynamic time warping score (nDTW): distance between the predicted and the ground-truth paths [43], oracle success rate (OSR): the ratio of reaching within 3 meters to the target, success rate (SR): the ratio of stopping within 3 meters to the target, and success weighted by the normalized inverse of the path length (SPL) [2].

### 4.2. Ablation Study

Ablation experiments in Table 1 reveal the influence of different pre-training objectives. We establish our Baseline by using a frozen CLIP-ViT-B/16 to encode RGB and depth separately, followed by a trainable fusion layer to merge the two features, which is a strong baseline that can achieve better results than the previous best encoders (see CLIP-ViT+Depth in Table 4).

Results of Model#1 indicate that employing angular off-set prediction as the only task has a devastating effect on the encoder; in fact, we found that $L_{\theta}$ only oscillates if it is minimized alone, leading to features that are not useful in downstream tasks (see details in Appendix). Although learning $L_{\theta}$ with $L_d$ or $L_c$ can improve the performance in both $A_{High}$ and $A_{Low}$ (Model#5, Model#6), removing the task from Model#7 will not cause a noticeable difference (Model#4). Meanwhile, learning explorable distance prediction (Model#2, Model#5) is not effective in $A_{High}$ scenario because agents with $A_{High}$ apply pre-defined way-points on open space, which means, finding explorable directions and avoid obstacle are not necessary. However, Model#2, Model#4 and Model#6 suggest that learning $L_d$ with $L_c$ or $L_{\theta}$ will boost the results in $A_{Low}$ and effectively reduce the collision rate during navigation.

On the other hand, by comparing the Baseline, Model#3, Model#6, and Model#7, we can see that our proposed Ego²-Map contrastive learning has the largest impact on R2R-CE. Solely learning $L_c$ improves the SR and SPL absolutely by 3.10% and 3.34% in $A_{High}$, as well as 2.01% and 2.02% in $A_{Low}$, while removing $L_c$ from Model#7 dramatically lowers the results in both settings. Meanwhile, we found that Ego²-Map learning can also help avoid collision, implying the useful spatial information carried by the map. In the following experiments, all three losses are applied to pre-train the visual encoders.

### 4.3. Discussion and Analysis

**How much does the quantity of visual data matter?** In Table 2, we compare the effect of sampling a different quantity of data for pre-training our visual encoder. Results show that the agent’s performance in R2R-CE drops as the amount of data decreases, and reducing the number of environments has a stronger impact on the results (-1.08% SPL for 50% samples vs. -3.87% SPL for 50% envs with $A_{High}$), suggesting the importance of having abundant scenes and structures in learning visual represen-

---

<table>
<thead>
<tr>
<th>Models</th>
<th>Pre-Training Objectives</th>
<th>R2R-CE Val-Unseen ($A_{High}$)</th>
<th>R2R-CE Val-Unseen ($A_{Low}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Pre-Training Objectives</td>
<td>NE</td>
<td>nDTW</td>
</tr>
<tr>
<td>Baseline</td>
<td>✓ ✓ ✓ ✓ ✓</td>
<td>4.98</td>
<td>57.26</td>
</tr>
<tr>
<td>1</td>
<td>✓ ✓ ✓ ✓ ✓</td>
<td>8.33</td>
<td>52.02</td>
</tr>
<tr>
<td>2</td>
<td>✓ ✓ ✓ ✓ ✓</td>
<td>5.83</td>
<td>51.25</td>
</tr>
<tr>
<td>3</td>
<td>✓ ✓ ✓ ✓ ✓</td>
<td>5.89</td>
<td>58.49</td>
</tr>
<tr>
<td>4</td>
<td>✓ ✓ ✓ ✓ ✓</td>
<td>5.03</td>
<td>58.86</td>
</tr>
<tr>
<td>5</td>
<td>✓ ✓ ✓ ✓ ✓</td>
<td>5.02</td>
<td>59.62</td>
</tr>
<tr>
<td>6</td>
<td>✓ ✓ ✓ ✓ ✓</td>
<td>5.79</td>
<td>52.93</td>
</tr>
<tr>
<td>7</td>
<td>✓ ✓ ✓ ✓ ✓</td>
<td>4.94</td>
<td>59.67</td>
</tr>
</tbody>
</table>

Table 1: Ablation of Ego²-Map learning and influence of angular and explorable objectives on the R2R-CE tasks. Checkmarks indicate applying the corresponding objectives. Baseline applies a pre-trained CLIP-ViT-B/16 model to encode RGB and depth separately. Collision measures the percentage of forward steps which the agent collided with an obstacle; we only consider this metric for $A_{Low}$ since agents in $A_{High}$ apply a waypoint predictor [41] to obtain navigable positions.
Effect of $\mathcal{L}$ on a different encoder We further evaluate our proposed Ego$^2$-Map objective in pre-training a different visual encoder (Table 4). Here we consider the previous state-of-the-art encoder CLIP-ViT+B/16 and a ResNet-50 trained for PointNav [94] to encode RGB and depth, respectively.

Information in Ego$^2$-Map contrastive learning In Table 5, we ablate the information applied in the Ego$^2$-Map contrastive learning. Specifically, we either remove the object semantics by masking their colored segmentations with the same color, or remove explorable regions by masking open space and void with the same color (see unmasked maps in Figure 2), or remove the target view from the view pairs to create ambiguity in agent’s transition. The results show that learning Ego$^2$-Map without the semantic clues or open space in the maps, or without a specified target greatly damages agent performance in $A_{High}$, reflecting the importance of including this information. We also find that the agent with $A_{Low}$ is less sensitive to method variations in pre-training; the remaining spatial or object information on a map is still beneficial to enhance the visual representation for supporting $A_{Low}$ navigation, which could be the reason why angular offset and exploratory distance predictions do not show a consistent benefit in Table 1.

4.4. Comparison to Previous Methods

Advantages of Visual Representations (R2R-CE) In Table 6, we compare our method of applying Ego$^2$-Map RGBD encoder to the CWP-VLN$^+$BERT agent [41] (same model as Model#7 in Table 1) with previous approaches.
using $A_{tri}$ on the R2R-CE testing split$^4$. Results show
that our proposed Ego$^2$-Map learning brings significant im-
provement for all metrics, achieving a 47% SR (+3%) and a
41% SPL (+4%) over the previous best [50]. In addition,
Table 7 establishes a fair comparison with methods using
$A_{Low}$. We can see that Ego$^2$-Map features greatly boost the
result of the base agent CWP-VLN$^\odot$BERT [41] (23% to
30% SR), and achieve a SPL comparable to recent mapping-
based methods such as CM2 [33] and WS-MGMap [14].
Note that using Ego$^2$-Map representations does not conflict
with online mapping; while we only experiment with non-
mapping-based agents, we believe the features hold great
potential to facilitate modeling between views and maps in
mapping-based models.

### Learning Methods (ObjNav)

We further compare the ef-
fected of visual representation learning methods by apply-
ning our visual encoder on ObjNav [4] as in the recent ap-
proaches [46, 96] (Table 8). All methods in the table adopt
a simple pipeline as in the baseline model [94], which feeds
the concatenation of visual features, GPS+Compass encod-
ings, and the encodings of previous action to a GRU [19],
then, uses a fully-connected layer to predict an action from
the updated agent’s state. Briefly, EmbCLIP directly uses
the pre-trained CLIP-ResNet50 [72] to encode the RGB
inputs, EmbCLIP-ViT+Depth applies the CLIP-ViT-B/16 to
encode RGB and an extra depth net [82] pre-trained on Gibson
[95] to encode the depth inputs. OVRL pre-trains the
ResNet encoder with self-distillation method DINO [6], in
which a student network is trained to match the output of
a teacher network. Moreover, OVRL applies the Omnidata
Starter Dataset (OSD)$^5$ [28], which is much larger and more
diverse than the data for our Ego$^2$-Map learning (only uses
the HM3D [74] subset). ObjNav measures the same metrics
as R2R-CE, first, we can see that an agent using Ego$^2$-Map
features greatly improves over the baseline and the Emb-
CLIP. Compared to OVRL, despite the method applies OSD
for pre-training and fine-tunes the network end-to-end with
human demonstrations, our method obtains a better success
rate (+0.4%) and much higher SPL (+3.2%). Note that, al-
though not directly comparable, as reported in the OVRL,
the SPL on ImageNav [63] without fine-tuning the visual
encoder will drop drastically from 26.9% to 17.0%, whereas
all our experiments keep the visual encoder frozen during
navigation. These results suggest that Ego$^2$-Map represen-
tations are generalizable to different navigation tasks and
provide more robust visual representations.

---

$^4$R2R-CE Challenge Leaderboard: https://eval.ai/web/challenges/challenge-
page/719/leaderboard/1966

$^5$OSD [28] contains approximately 14.5 million images rendered from
diverse 3D environments, including Replica [88], Replica+GSO [26], Hyp-
ersim [79], Taskonomy [100], BlendedMVG [97] and HM3D [74].

<table>
<thead>
<tr>
<th>Methods ((A_{tri}))</th>
<th>R2R-CE Test-Unseen</th>
<th>TL</th>
<th>NE</th>
<th>OSR</th>
<th>SR</th>
<th>SPL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Waypoint Models [49]</td>
<td></td>
<td>8.02</td>
<td>6.65</td>
<td>37</td>
<td>32</td>
<td>30</td>
</tr>
<tr>
<td>CWP-CMA [41]</td>
<td></td>
<td>11.85</td>
<td>6.30</td>
<td>49</td>
<td>38</td>
<td>33</td>
</tr>
<tr>
<td>CWP-VLN$^\odot$BERT [41]</td>
<td></td>
<td>13.31</td>
<td>5.89</td>
<td>51</td>
<td>42</td>
<td>36</td>
</tr>
<tr>
<td>Sim2Sim [50]</td>
<td></td>
<td>11.43</td>
<td>6.17</td>
<td>52</td>
<td>44</td>
<td>37</td>
</tr>
<tr>
<td>Ego$^2$-Map+CWP-VLN$^\odot$BERT (ours)</td>
<td></td>
<td>13.05</td>
<td>5.54</td>
<td>56</td>
<td>47</td>
<td>41</td>
</tr>
</tbody>
</table>

Table 6: Comparison of agent performance on R2R-CE test server. All methods use high-level action space (\(A_{tri}\)).

<table>
<thead>
<tr>
<th>Methods ((A_{Low}))</th>
<th>R2R-CE Val-Unseen</th>
<th>TL</th>
<th>NE</th>
<th>OSR</th>
<th>SR</th>
<th>SPL</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMA+PM+DAVAug [52]</td>
<td>8.64</td>
<td>7.77</td>
<td>40</td>
<td>32</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>SASRA [44]</td>
<td>7.89</td>
<td>8.32</td>
<td>24</td>
<td>22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LAW [77]</td>
<td>8.89</td>
<td>6.83</td>
<td>44</td>
<td>35</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>CWP-CMA [41]</td>
<td>8.22</td>
<td>7.54</td>
<td>27</td>
<td>25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CWP-VLN$^\odot$BERT [41]</td>
<td>7.42</td>
<td>7.66</td>
<td>23</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CM2 [33]</td>
<td>11.54</td>
<td>7.02</td>
<td>42</td>
<td>34</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>WS-MGMap (SemMap only) [14]</td>
<td>10.89</td>
<td>6.80</td>
<td>42</td>
<td>33</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>Ego$^2$-Map+CWP-VLN$^\odot$BERT (ours)</td>
<td>8.03</td>
<td>7.25</td>
<td>37</td>
<td>30</td>
<td>29</td>
<td></td>
</tr>
</tbody>
</table>

Table 7: Comparison of agents with low-level action space (\(A_{Low}\)) in R2R-CE Val-Unseen. †: mapping-based methods. $^\odot$: 4% of data is removed but the comparison is still valid due to the large performance gap.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Pre-Training Dataset</th>
<th>ObjNav MP3D Val</th>
<th>NE</th>
<th>SR</th>
<th>SPL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline [92]</td>
<td></td>
<td>6.90</td>
<td>8.00</td>
<td>1.8</td>
<td></td>
</tr>
<tr>
<td>EmbCLIP [46]</td>
<td>WebImageText</td>
<td>5.26</td>
<td>20.9</td>
<td>8.3</td>
<td></td>
</tr>
<tr>
<td>EmbCLIP+ViT+Depth</td>
<td>WebImageText+Gibson</td>
<td>4.90</td>
<td>23.3</td>
<td>8.6</td>
<td></td>
</tr>
<tr>
<td>OVRL no pretrain [76]</td>
<td>$^\odot$</td>
<td>24.2</td>
<td>5.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OVRL [96]</td>
<td>OSR</td>
<td>28.6</td>
<td>7.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ego$^2$-Map+Baseline (ours)</td>
<td>HM3D</td>
<td>5.17</td>
<td>29.0</td>
<td>10.6</td>
<td></td>
</tr>
</tbody>
</table>

Table 8: Comparison on pre-trained visual encoders for Ob-
jNav. † indicates the visual encoder is tuned end-to-end
with behavior cloning on 40k human demonstrations col-
lected by Habitat-Web [76], while the others freeze the vi-
sual encoder during navigation and only train the agent with
DD-PPo [94] on the original data. $^\odot$ Results obtained by re-
evaluating the officially released best model checkpoint.

### 5. Conclusion

In this paper, we introduce a novel method of learning
navigational visual representations with contrastive learn-
ing between egocentric views pairs and top-down semantic
maps (Ego$^2$-Map). The method transfers the compact se-
matic and spatial information carried by a map to the ego-
centric representations, which greatly facilitates the agent’s
visual perception. Experiments show that Ego$^2$-Map fea-
tures greatly improve the downstream navigation, such as
ObjNav and VLN, and demonstrate generalization potential
to different visual backbones. We believe the Ego$^2$-Map
contrastive learning proposes a new direction of visual rep-
resentation learning for navigation and provides the possi-
bility of better modeling the correspondence between views
and maps, which can further benefit agent’s planning and
action. Note that our work also produces a potentially ef-

fective map encoder whose full capability is worth investigating in future work.

Limitations The need to build semantic maps to enable the Ego²-Map contrastive learning is an inevitable cost of this method; compared to other self-supervised visual representation learning approaches, Ego²-Map requires either the semantic annotations of scenes, or traversable environments and a generalizable semantic map constructor. As a result, the data collection could be much harder. However, we also witnessed an increasing number of interactive 3D scenes being built in recent years with dense semantic annotations [22, 74, 82, 88, 95], which can facilitate scaling up our Ego²-Map learning in the future.

References

[20] Yuchen Cui, Scott Nickum, Abhinav Gupta, Vikash Kumar, and Aravind Rajeswaran. Can foundation models per-


