Monocular Camera-Based Point-Goal Navigation by Learning Depth Channel and Cross-Modality Pyramid Fusion

Tianqi Tang[1]*, **Heming Du**[2]*, **Xin Yu**[1], **Yi Yang**[1]†
[1]ReLER, AAII, University of Technology Sydney, Australia
[2]Australian National University
Tang@student.uts.edu.au, heming.du@anu.edu.au, {xin.yu, yi.yang}@uts.edu.au

Abstract

For a monocular camera-based navigation system, if we could effectively explore scene geometric cues from RGB images, the geometry information will significantly facilitate the efficiency of the navigation system. Motivated by this, we propose a highly efficient point-goal navigation framework, dubbed Geo-Nav. In a nutshell, Geo-Nav consists of two parts: a visual perception part and a navigation part. In the visual perception part, we firstly propose a Self-supervised Depth Estimation network (SDE) specially tailored for the monocular camera-based navigation agent. SDE learns a mapping from an RGB input image to its corresponding depth image by exploring scene geometric constraints in a self-consistency manner. Then, in order to achieve a representative visual representation from the RGB inputs and learned depth images, we propose a Cross-modality Pyramid Fusion module (CPF). Concretely, CPF computes a patch-wise cross-modality correlation between different modal features and exploits the correlation to fuse and enhance features at each scale. Thanks to the patch-wise nature of CPF, we can fuse feature maps at high resolution, allowing the visual network to perceive more image details. In the navigation part, the extracted visual representations are fed to a navigation policy network to learn how to map the visual representations to agent actions effectively. Extensive experiments on the Gibson benchmark demonstrate that Geo-Nav outperforms the state-of-the-art in terms of efficiency and effectiveness.

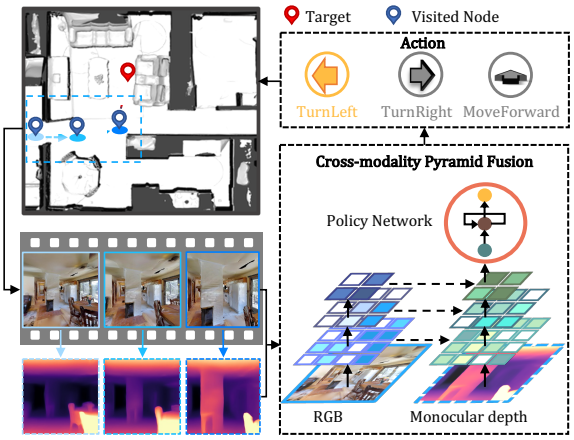

Figure 1: Illustration of Geo-Nav framework. An agent predicts monocular depth by Self-supervised Depth Estimation network (SDE). The agent applies Cross-modality Pyramid Fusion module (CPF) to merge features from the RGB image and estimated depth map into an informative visual representation. Then, the agent produces a navigation policy and selects an action, *e.g.*, `TurnLeft`. The red point represents the target position, and visited nodes are highlighted by blue points in one episode.

Introduction

Point-goal visual navigation aims to instruct an agent to move towards a target position based on RGB observations and directional indications. Apart from the indications, fully understanding scene geometry plays a critical role in navigation action reasoning. Particularly, sensing scene geometry significantly facilitates the efficiency of the navigation system, such as avoiding obstacles.

Recent works validate the importance of depth information in visual navigation. For example, previous methods (Chen et al. 2019; Wijmans et al. 2020) employ RGB-D images to learn expressive visual representations. Some navigation systems (Gordon et al. 2019; Sax et al. 2018) pre-train agents with auxiliary tasks, *e.g.*, depth prediction, to exploit geometric information. However, such navigation systems may suffer from severe performance degradation when depth information is not available. Unlike previous works that require ground-truth depth, we develop a monocular camera-based navigation framework named Geo-Nav, which can efficiently explore scene geometric cues without relying on ground-truth depth information.

Geo-Nav consists of two parts: a visual perception part and a navigation part. In the visual perception part, we firstly introduce a Self-supervised Depth Estimation network (SDE) to comprehend scene geometric structure from RGB observations by estimating dense depth maps of scenes. To be specific, SDE learns to map monocular RGB images to depth maps from the visual observation differences between consecutive frames. We adopt an encoder-decoder architecture to estimate scene depth by satisfying

*These authors contributed equally.
†Corresponding author

the geometric constraints between consecutive frames and enforcing estimated depth scales to be consistent. As we aim to leverage the inherent geometric information to learn depth maps during navigation, pure-rotation image pairs are not suitable for SDE to learn the depth. Therefore, we sample pure-translation movements from historical trajectories and then train SDE with the sampled image pairs.

In order to generate expressive visual representations from RGB and depth images for the navigation policy network, we design a Cross-modality Pyramid Fusion module (CPF). Specifically, we employ a ResNet18 (He et al. 2016) to extract image contextual cues from RGB observations and an encoder network to extract scene geometry from estimated depth maps. Recall that the scene geometry and image contextual information are two distinct modalities. Thus, it is less effective to apply simple late-fusion methods, such as fusion by addition. To tackle this issue, CPF is designed to fuse representations of RGB and depth hierarchically. Concretely, CPF first computes a patch-wise cross-modality correlation between these two modal features and then fuses these features based on the correlation at multiple scales separately. The embedding patches can be viewed as the same way as tokens in a natural language processing (NLP) application (Vaswani et al. 2017). CPF attends the depth embedding tokens to RGB features with the corresponding positional embeddings to construct the patch-wise relationships between two modalities. Furthermore, since CPF measures correlation in a patch-wise manner, CPF can be applied to the high-resolution feature maps and preserves rich spatial details. In this way, we obtain more informative visual representations.

In the navigation part, agents embed directional signals, previous actions and visual representations into the navigation policy network. We adopt a Long Short-Term Memory network (LSTM) to generate navigation actions. To speed up the training procedure of the navigation network, we warm up the navigation policy network via imitation learning (IL) with expert demonstrations. Then, we adopt the Proximal Policy Optimization (PPO) paradigm (Schulman et al. 2017) to learn a general navigation policy. Experiments demonstrate that Geo-Nav outperforms the state-of-the-art methods in the widely-used Gibson benchmark (Xia et al. 2018). Noticeably, the entire network is trained $6\times$ faster than DD-PPO (Wijmans et al. 2020) when achieving similar performance. This also indicates that Geo-Nav effectively exploits geometric cues and thus improves training efficiency.

Overall, our contributions are summarized as follows:

- We propose a scene-geometry driven point-goal RGB image based visual navigation framework, dubbed as Geo-Nav. It is designed to integrate image cues and scene geometry explored from the consecutive frames into an expressive representation.
- We introduce a scale-consistent Self-supervised Depth Estimation network (SDE) specially tailored for visual navigation. To the best of our knowledge, Geo-Nav is the first attempt to exploit self-supervised depth from a monocular RGB camera in point-goal navigation tasks.
- We develop a Cross-modality Pyramid Fusion module (CPF) to associate scene geometry with image contents, thus significantly alleviating depth scale ambiguity and enhancing visual feature representations.
- Extensive experiments demonstrate that Geo-Nav outperforms the state-of-the-art methods on the Gibson benchmark (Xia et al. 2018) in terms of efficiency and effectiveness. Moreover, due to our effective exploitation of scene geometry, Geo-Nav also achieves better training efficiency than the state-of-the-art.

Related Work

Recent reinforcement learning-based navigation works mainly focus on steering an agent to target locations in unknown environments, while traditional methods construct maps of environments to find the shortest path for navigation. Here, we briefly review the most relevant literature.

SLAM-based navigation methods (Cadena et al. 2016; Durrant-Whyte and Bailey 2006; Garcia-Fidalgo and Ortiz 2015) generally consist of three steps: building a map, localizing the agent in the map, and designing an optimal path based on the map. Prevalent SLAM systems (Davison et al. 2007; Klein and Murray 2007; Mur-Artal, Montiel, and Tardos 2015) are based on hand-crafted or learned features (Tian et al. 2019; Yu et al. 2019, 2020; Li et al. 2020, 2021). However, traditional feature-based SLAM might fail when a scene is textureless and may suffer drifting problems.

Reinforcement Learning (RL) based methods employ expressive representations and effective policies to maximize the cumulative reward. The pioneering work (Zhu et al. 2017) proposes a target-driven based navigation model and a new simulated environment for obtaining enough training samples. Following works (Gordon et al. 2019; Savva et al. 2019) exploit an end-to-end reinforcement learning pipeline with an RGB-D camera, and they demonstrate an RGB-D camera is more effective than a monocular RGB camera. Gupta et al. (2017) construct a joint differentiable mapper and planner for mimicking the classical SLAM, but their approach requires depth information to obtain the optimal actions for an agent.

Shen et al. (2019) leverage 25 diverse vision tasks (*e.g.*, supervised depth predictions) to improve the robustness of navigation policy. Recent works (Du, Yu, and Zheng 2020, 2021; Tang et al. 2021) validate that efficient exploration policies facilitate downstream navigation training. In particular, Chaplot et al. (2020a) require ground-truth depth as supervision signals, while Chaplot et al. (2020b) need a topological map of the environment in advance for training. However, these types of information are not available to realistic agents during training. Note that most methods rely on depth sensors or prerequisite depth maps, while Geo-Nav is designed to learn scene depth directly from the historical trajectories without requiring ground-truth depth information.

Meanwhile, several works (Gordon et al. 2019; Ye et al. 2020; Zhu et al. 2020) leverage self-supervised auxiliary tasks, such as predicting next frames or next actions. However, those methods do not *explicitly* exploit scene geometric information for navigation. Although some works (Wagstaff and Kelly 2021; Zhou et al. 2017; Godard et al. 2019; Bian

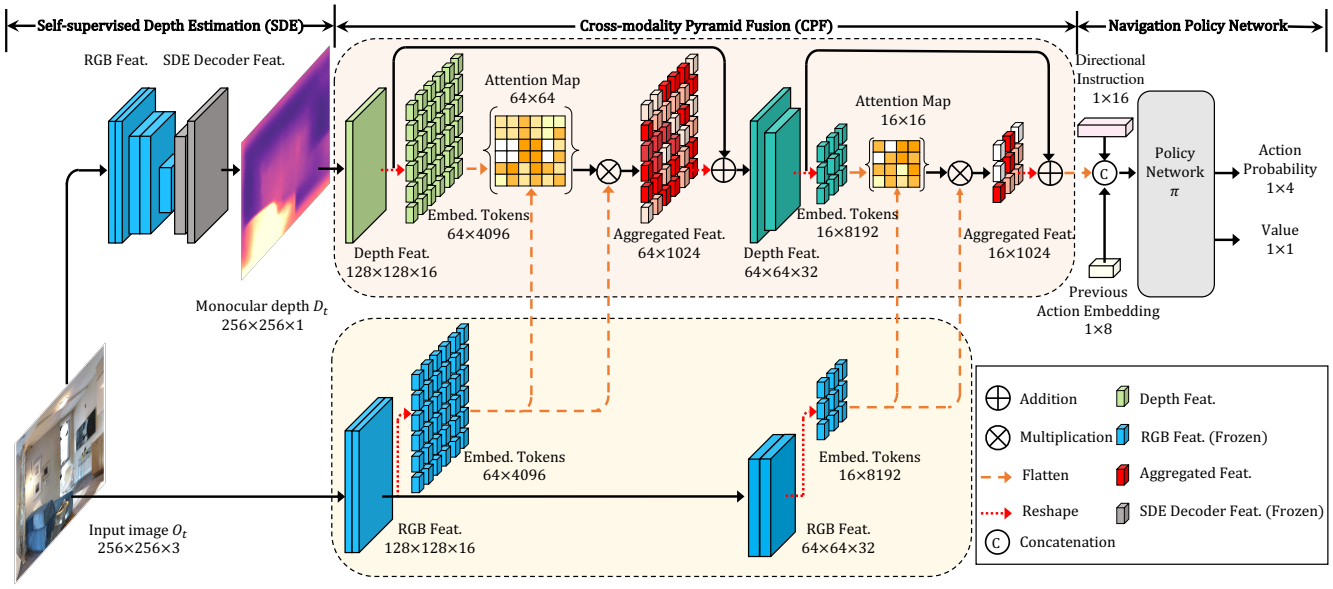

Figure 2: Overview of Geo-Nav framework. Geo-Nav involves two parts: (i) visual perception encoder and (ii) a standard policy network. In the visual perception encoder, We employ two branches to extract the features. SDE infers depth from the RGB observations. Meanwhile, CPF computes patch-wise attention maps between RGB representations and depth features and then aggregates weighted features at different scales. Then, the policy network encodes the informative visual representations, directional signals and previous actions into the policy network.

et al. 2019; Godard, Mac Aodha, and Brostow 2017; Zhan et al. 2018; Shi, Li, and Yu 2021) propose self-supervised depth estimation methods, directly using these methods in navigation will suffer depth scale ambiguity and lead to inferior navigation performance. Unlike the work of Wagstaff and Kelly (2021) that adopts scale consistency in one direction, we constrain the scale consistency in two directions, which helps Geo-Nav to produce highly scale-consistent depth estimation.

Proposed Method

Our navigation framework Geo-Nav is composed of a visual perception part and a navigation part. In the visual perception part, we introduce a scale-consistent Self-supervised Depth Estimation network (SDE) to explore scene geometric information and then propose a Cross-modality Pyramid Fusion module (CPF) to integrate the estimated depth and image contextual cues into expressive visual representations. The goal of the navigation part is to learn an effective and efficient navigation strategy based on visual representations.

Task Definition

The objective of monocular camera-based point-goal navigation is to steer an agent towards a target point with RGB observations and coarse directional indications. To be specific, at the timestamp t, an agent obtains an RGB observation O_t from its monocular camera. Meanwhile, the agent receives a directional indication I_t, comprising the distance d_t and orientation o_t pointing to the end node n_e (*i.e.* target).

The agent extracts visual representations and learns to approximate an optimal navigation policy $\pi(a_t|O_t, I_t, h_{t-1})$, where h_{t-1} stands for the previous hidden state, and a_t represents the distribution of actions. Then, the agent selects the action with the highest probability for navigation. The action space is composed of 4 distinct candidates, including `MoveForward`, `TurnLeft`, `TurnRight` and `Stop`. More specifically, the forward step size is 0.25 meters, and the angle of `TurnLeft/TurnRight` is 10°.

With maximum steps (*e.g.*, 500 steps), an episode is regarded as a success when the position of the agent is within 0.2 meters of the target node n_e. Otherwise, it will be considered a failure episode.

Scale-consistent Self-supervised Depth Estimation

To infer depth information only from RGB observations, we introduce a Self-supervised Depth Estimation network (SDE), as illustrated in Figure 3, into the Geo-Nav framework. We first train SDE with image pairs from agents' trajectories. Specifically, SDE estimates both depth maps $(D_t, D_{t'})$ and pose $P_{t\to t'}$ between two consecutive images $(O_t, O_{t'})$. Given D_t, $P_{t\to t'}$ and the camera intrinsic matrix $\mathbb{K}$, we calculate the projection function (Zhou et al. 2017) to transform the pixel coordinates between images. Following Jaderberg et al. (2015), we warp the source image $O_{t'}$ to the target image $O_{t'\to t}$ based on the projection function. Here, we utilize three self-supervised losses in training.

Using the brightness constancy priors (Bruhn, Weickert, and Schnörr 2005), we introduce the photometric loss to

minimize the differences between O_t and $O_{t'\to t}$, as follows:

$$\mathcal{L}_p = \frac{1}{|\mathbb{V}|}\sum_{\mathbb{V}} \| O_t - O_{t'\to t} \|_1 , \tag{1}$$

where $\mathbb{V}$ indicates the set of overlapped pixels between $O_{t'}$ and O_t, and $|\mathbb{V}|$ is the total number of pixels. Moreover, to eliminate the effects of illumination changes, we adopt the structural similarity index measure (SSIM) loss. We add the SSIM loss to Eq. (1) and measure the similarity between the target image and the warped one. Following this, the photometric loss is formulated as:

$$\mathcal{L}'_p = (1-\lambda)\mathcal{L}_p + \frac{\lambda}{2}(1 - \mathit{SSIM}(O_t, O_{t'\to t})), \tag{2}$$

where λ is a trade-off weight. We set $\lambda = 0.85$ for all the experiments.

Since rendered images may contain low-texture regions, the photometric loss is not effective in these regions. We leverage an edge-aware smoothness loss to generate smooth depth, expressed as:

$$\mathcal{L}_s = |\nabla_x D_t^*| e^{-|\nabla_x O_t|} + |\nabla_y D_t^*| e^{-|\nabla_y O_t|}, \tag{3}$$

where ∇ represents the first-order derivative along spatial directions (Rother, Kolmogorov, and Blake 2004), and D_t^* is the normalized inverse depth.

As stated in Godard et al. (2019), self-supervised monocular depth estimation often suffers scale ambiguity. Training such image pairs cannot guarantee that the output depth is scale-consistent. In the navigation task, the unstable depth estimation could mislead the policy, causing navigation failure. To obtain scale-consistent predictions over the entire sequences, we employ a bidirectional geometry consistency regularization, defined as:

$$\mathcal{L}_{GC} = \frac{|D_{t\to t'} - D_{t'}|}{D_{t'}} + \frac{|D_{t'\to t} - D_t|}{D_t}, \tag{4}$$

where $D_{t\to t'}$ represents a warped depth map from the predicted depth map D_t. Consequently, the overall objective function is formulated as below:

$$\mathcal{L}_{SDE} = \lambda_1 \mathcal{L}'_p + \lambda_2 \mathcal{L}_s + \lambda_3 \mathcal{L}_{GC}, \tag{5}$$

where $\lambda_1, \lambda_2, \lambda_3$ stand for the trade-off weights. We empirically set $\lambda_1 = 1$, $\lambda_2 = 0.1$ and $\lambda_3 = 0.5$ to achieve good quality of depth maps. Thanks to the scale consistency regularization, we can obtain scale-consistent depth predictions, thus improving the navigation efficiency.

Cross-modality Pyramid Fusion

We design a Cross-modality Pyramid Fusion module (CPF) to integrate the features of RGB images and depth, as shown in Figure 4. CPF has two main components: a cross-modality attention module (CA) and a multi-scale fusion strategy (MSF). The goal of CA is to explore correlations between depth and RGB representations.

In order to leverage image details at high resolution, we adopt a patch-wise attention mechanism when fusing both modalities. Specifically, we split features from both modalities into a sequence of patches with a patch size of 16×16

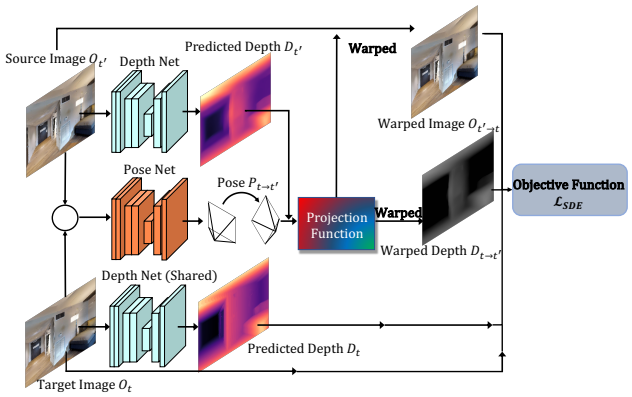

Figure 3: Illustration of Self-supervised Depth Estimation network (SDE).

pixels, and thus the feature dimension of each patch is 16×16×C, where C is the number of channels.

Then, a linear embedding layer is applied to these features, projecting the outputs into the target dimension (*i.e.*, 1024). Meanwhile, we use learnable positional embedding to represent the relative positions of the patches. We add learnable positional embedding E_{pos} to the corresponding flattened patches. Those modified patches are then passed to CA. CA then fuses the RGB and depth information effectively in a memory-efficient way benefiting from the patch-wise nature of CA.

Considering RGB images contain detailed texture and depth images mainly contain coarse structure information, the depth embedding tokens are transformed into a query matrix Q, and the image embedding tokens are transformed into a key matrix K and a value matrix V. Then, agents compute the similarity between the key matrix K and the query matrix V to explore the patch-wise relationship across multiple modalities. Meanwhile, we utilize the multi-head architecture (Vaswani et al. 2017) to extract expressive information from different representation subspaces. The aggregated feature is then computed as a weighted sum of the values. Aggregated features are passed to feed-forward networks, which are formulated by linear transformations. The attention mechanism is represented as:

$$Q = [E_p^1; E_p^2; ...; E_p^N] + E_{pos}, \tag{6}$$

$$K, V = [\widetilde{E}_p^1; \widetilde{E}_p^2; ...; \widetilde{E}_p^N] + E_{pos}, \tag{7}$$

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d'_k}})V, \tag{8}$$

where E_{pos} is the positional embedding, and d'_k represents a scaling factor. E_p^i and $\widetilde{E}_p^i$ stand for the i-th depth embedding patch and RGB feature patch.

Furthermore, CPF allows multiple CAs at different scales to achieve expressive representations. Specifically, both RGB and depth feature extractors are divided into four stages with strides of $\{2, 4, 8, 16\}$. At each stage, we compute the patch-wise attention map and feed the weighted sum

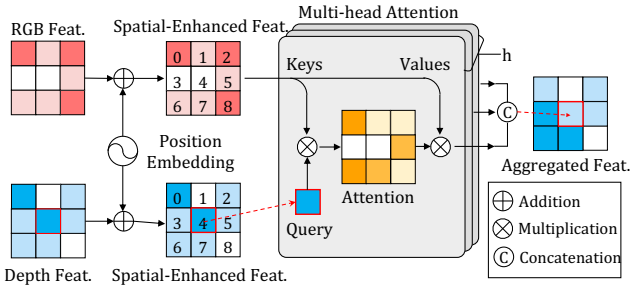

Figure 4: Illustration of Cross-modality Attention (CA). Both embedding tokens share the position embeddings. Here, we adopt the multi-head cross-modality attention.

of features into the next layer. The inputs and outputs of CA at each scale are connected by a skip connection and followed by a layer normalization, as seen in Figure 2. Since the outputs from CA do not match the shape of the previous 2D features, we provide a reshape layer to change the fused feature shape.

Navigation Policy Network

To demonstrate the expressiveness of visual representations, we only adopt standard PPO (Schulman et al. 2017) architecture for navigation policy learning. The policy network consists of a Long short-term memory network (LSTM), policy head and value head. Both heads are composed of the fully-connected layers and learn the mapping from hidden states to action distributions and values. The input of the policy network is the concatenation of visual representations, previous action embeddings, directional instruction embeddings and the previous hidden states.

We adopt Proximal Policy Optimization (PPO) technique (Schulman et al. 2017) to approximate the optimal navigation policy. The policy network produces two outputs, *i.e.*, policy and values. We sample the action from the predicted policy with the highest probability, as seen in Figure 2. Therefore, the agent switches to the next state and moves towards the target position.

Training Details

We first train SDE on the training data sampled from the Gibson (Xia et al. 2018) environment. To avoid noise from pure-rotation movements (Bian et al. 2020), we only collect the pure-translation movements from the environment. As suggested by Bian et al. (2020), the step size of the agent is set to $0.05m$ to ensure the similarities among consecutive images. We sample 1K image triplets for each scene, and there are in total 180K triplets for SDE training. We jointly train the depth net and the pose net, as shown in Figure 3. During inference, the pure-rotation movements will not affect the performance of SDE because SDE estimates depth from a single RGB image.

After training SDE, we optimize the navigation network, including the policy network and visual perception encoder. We employ two branches to extract features from RGB images and depth maps, respectively. The backbone in the

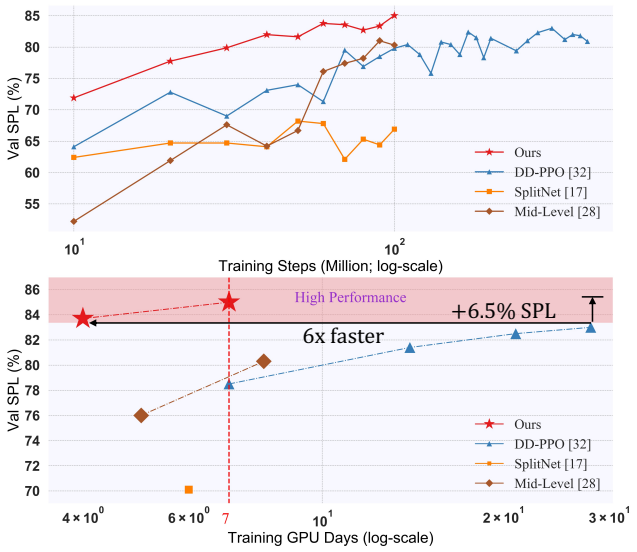

Figure 5: Top: SPL on the Validation set of Gibson within 100 million steps. Bottom: SPL vs. GPU training time. We observe that Geo-Nav uses $6\times$ fewer steps (*i.e.*, $6\times$ faster to be trained) than DD-PPO (Wijmans et al. 2020) when achieving similar performance.

RGB branch is a ResNet18 initialized with the weights of the trained SDE encoder. In training the policy network, we freeze the parameters of the image branches.We first warm up the agent via imitation learning (IL), where optimal navigation actions are provided to the navigation policy network as supervision. The agent learns the optimal action a_t^* at each step, formulated as $L_{IL} = \sum_t -a_t^* log(p_t(a_t))$, where a_t^* is the one-hot encoding of an optimal action. p_t is the distribution of action space at time step t.

Furthermore, we fine-tune the agent by a standard PPO algorithm (Schulman et al. 2017) to enhance the generalization of navigation policy. The reward function used to optimize the navigation network is defined as:

$$r_t = \begin{cases} \theta + d_{t-1} - d_t + \lambda_t, & \text{if reach the goal,} \\ d_{t-1} - d_t + \lambda_t, & \text{otherwise,} \end{cases} \quad (9)$$

where d_t is the geodesic distance to the target point at timestamp t, θ is a constant reward which will be activated when the agent reaches the target goal, and λ_t is a time penalty.

Experiments

Dataset

We train and evaluate agents on the platform Habitat (Savva et al. 2019), which provides agents with photo-realistic images through virtual simulations. All the experiments are conducted on the Gibson dataset (Xia et al. 2018). Gibson contains real-world indoor scenarios, including 5 million episodes in 72 indoor environments for training and 994 episodes in 14 unseen environments for evaluation. Each episode involves a starting position, a target position and the geodesic distance between the starting and target positions.

Method	SPL (%) ↑	Success (%) ↑
Random	$2.1_{\pm 0.01}$	$3.9_{\pm 0.34}$
SplitNet (Gordon et al. 2019)	$70.9_{\pm 0.36}$	$86.8_{\pm 0.97}$
Mid-Level (Sax et al. 2018)	$78.9_{\pm 2.01}$	$91.7_{\pm 0.23}$
DD-PPO (Wijmans et al. 2020)	$79.2_{\pm 0.22}$	$92.1_{\pm 0.25}$
DD-PPO$^{\pounds}$ (Wijmans et al. 2020)	$82.7_{\pm 0.18}$	$89.6_{\pm 0.06}$
Geo-Nav	$84.7_{\pm 0.15}$	$93.5_{\pm 0.30}$

Table 1: Navigation performance of the state-of-the-art and Geo-Nav.

Evaluation Metrics

Following Anderson et al. (2018), we employ Success weighted by Path Length (SPL) as one of evaluation metrics to measure the policy efficiency of the navigation framework. SPL is utilized to measure the deviation between the navigation path and the shortest one, formulated as $SPL = \frac{1}{N}\sum_{i=1}^{N} S_i \frac{l_i}{max(p_i,l_i)}$, where S_i represents a binary success indicator for episode i, p_i is the path length of the agent, and l_i is the shortest distance from the starting position to the target point. Meanwhile, we employ the success rate (Success) as another evaluation metric to demonstrate the effectiveness of a navigation system. The success rate is computed by $\frac{1}{N}\Sigma_1^N S_i$.

Experimental Details

During training SDE, we employ the Adam optimizer (Kingma and Ba 2014) with the batch size of 12 and a learning rate of 10^{-5}. Once SDE has been trained, it is fixed and only outputs depth maps. Then, we warm up the depth branch in the visual perception encoder and policy network via IL. We utilize Adam optimizer with a learning rate of 2.5×10^{-4} to train the proposed agent within 10 million steps. Finally, we fine-tune the agent and employ a lower learning rate of 10^{-4} for RL. To maintain navigation performance, we freeze the weights of the visual perception network during training IL and RL. Additionally, we train Geo-Nav with 100 million training steps on 1 Nvidia V100 GPU for seven days.

Compared Methods

Baseline (Res18-Baseline). Baseline extracts fixed RGB features through a pre-trained ResNet18 and trains the policy network based on the features for navigation.

Random policy (Random). The random policy chooses an action randomly from a uniform distribution.

SplitNet (Gordon et al. 2019). SplitNet adopts a multi-stage training strategy based on RGB images and other environment information (*e.g.*, depth).

Mid-Level (Sax et al. 2018). Mid-Level adopts a deeper ResNet50 backbone and also introduces a set of auxiliary tasks to improve visual representations. We train Mid-Level with the proposed max-coverage feature set.

DD-PPO (Wijmans et al. 2020). DD-PPO employs a much deeper visual encoder, *i.e.*, SE-ResNeXt101, and spends 180 GPU-days[1] on training. Due to the fact that the RGB-based DD-PPO utilizes additional data from Matterport3D (Chang et al. 2017), we train DD-PPO with only the Gibson dataset and evaluate the model that has been trained with 100 million steps for fair comparisons.

Method	SPL (%) ↑	Success (%) ↑
Blind	$72.5_{\pm 0.04}$	$86.8_{\pm 0.15}$
Res18-Baseline	$79.4_{\pm 0.12}$	$87.8_{\pm 1.68}$
Res18-Baseline ‡	$71.4_{\pm 0.18}$	$82.3_{\pm 0.56}$
Res18-Baseline w/ SDE	$80.0_{\pm 0.07}$	$91.4_{\pm 0.06}$
Geo-Nav $\dotplus$	$85.2_{\pm 0.15}$	$94.7_{\pm 0.13}$
Geo-Nav $\nmid$	$84.6_{\pm 0.05}$	$94.0_{\pm 0.16}$
Geo-Nav ‡	$78.2_{\pm 0.68}$	$88.9_{\pm 0.21}$
Geo-Nav w/o $\mathcal{L}_{GC}$	$68.9_{\pm 0.04}$	$85.3_{\pm 0.32}$
Geo-Nav	$84.7_{\pm 0.15}$	$93.5_{\pm 0.30}$

Table 2: The impacts of SDE on navigation performance. ‡ represents utilizing ImageNet pre-trained weights instead of proposed SDE pre-trained weights for image feature extractor. $\dotplus$ indicates adopting ground-truth depth instead of the estimated depth map. $\nmid$ stands for changing the self-supervised depth estimation method to supervised depth estimation method (Bhat, Alhashim, and Wonka 2020).

Method	Error↓			Accuracy↑		
	AbsRel	RMS	RMSlog	< 1.25	$< 1.25^2$	$< 1.25^3$
SDE	0.134	0.643	0.062	0.810	0.932	0.974
SDE w/o $\mathcal{L}_{GC}$	3.195	5.165	0.502	0.205	0.341	0.430

Table 3: Depth estimation performance of SDE.

Navigation Results

Comparison with State-of-the-art. As indicated in Table 1, Geo-Nav outperforms state-of-the-art methods (*e.g.*, SplitNet (Gordon et al. 2019), Mid-Level (Sax et al. 2018) and DD-PPO (Wijmans et al. 2020)) by a large margin on both SPL and success rate respectively. Both SplitNet and Mid-Level encode the image cues and ground-truth depth information implicitly by leveraging auxiliary tasks (*e.g.*, supervised depth estimation). Here, SplitNet also applies the same warm-up routine as Geo-Nav. In contrast, Geo-Nav infers scale-consistent depth maps from consecutive RGB images via SDE. Benefiting from SDE and CPF, Geo-Nav achieves explicit scene geometric information for navigation and thus improves navigation performance.

For fair comparisons, we train the original DD-PPO with only the Gibson dataset and 100 million training steps. We observe that Geo-Nav achieves SPL of 84.7%, while DD-PPO achieves SPL of 79.2% within the same training steps. The results imply that Geo-Nav is more training-efficient by fully exploring geometric cues. Since the original DD-PPO employs a much deeper backbone SE-ResNext101, we replace it with the backbone ResNet18, marked as DD-PPO$^{\pounds}$.

[1]A single V100 GPU runs 180 days to complete a task.

Method	SPL (%) ↑	Success (%) ↑
Res18-Baseline	$79.4_{\pm 0.12}$	$87.8_{\pm 1.68}$
Res18-Baseline w/ SDE	$80.0_{\pm 0.07}$	$91.4_{\pm 0.06}$
Geo-Nav w/o MSF	$82.9_{\pm 0.18}$	$92.0_{\pm 0.23}$
Geo-Nav w/o MSF §	$80.9_{\pm 0.11}$	$90.8_{\pm 0.06}$
Geo-Nav w/o CA	$81.5_{\pm 0.12}$	$91.8_{\pm 0.15}$
Geo-Nav †	$82.4_{\pm 0.06}$	$91.1_{\pm 0.15}$
Geo-Nav	$84.7_{\pm 0.15}$	$93.5_{\pm 0.30}$

Table 4: The impacts of different fusion mechanisms, including the multi-scale fusion (MSF) and cross-modality attention (CA). § represents a vanilla single-scale self-attention. † stands for opposite asymmetric design of attention.

Meanwhile, we warm up DD-PPO$^{\pounds}$ via IL to make a fair comparison. As seen in Table 1, Geo-Nav achieves better performance than DD-PPO$^{\pounds}$ on both the SPL and Success.

Furthermore, we validate that Geo-Nav is more training-efficient than other methods. In Figure 5, the top figure shows Geo-Nav achieves the best SPL of 85% and obvious performance gain during the training procedure. The bottom figure demonstrates Geo-Nav achieves the SPL of 83.6% using only four GPU-days, while DD-PPO requires 27 GPU-days with the SPL of 82.5%[2]. Thus, training Geo-Nav is 6 $\times$ faster than DD-PPO with similar performance.

Ablation Study

Impacts of SDE. To analyze the importance of SDE, we compare Res18-Baseline w/ SDE with Res18-Baseline in Table 2. Res18-Baseline w/ SDE fuses image features and depth features by addition. The comparison suggests that SDE improves the effectiveness of agents by introducing geometric cues. We also compare with a Blind model to demonstrate the impact of depth maps. The Blind model only extracts the descriptors from the depth branch and ignores the image contextual information. The result indicates that scene geometric information is very important in navigation. Furthermore, to demonstrate the importance of the scale-consistency regularization in SDE, we remove the regularization during training, marked as Geo-Nav w/o $\mathcal{L}_{GC}$. As indicated in Table 2, Geo-Nav attains better performance due to more consistent depth predictions. Moreover, we report the performance of SDE w/o $\mathcal{L}_{GC}$, which excludes the scale-consistency regularization, in Table 3. We observe that SDE w/o $\mathcal{L}_{GC}$ produces inconsistent depth maps, which hinder an agent in producing correct behaviors.

Impacts of CPF. To analyze the impacts of CA and MSF, we compare the ablated versions of our method in Table 4. We replace CA with the addition operation to evaluate the enhancement of MSF on Res18-Baseline. As shown in Table 4, MSF improves the agent slightly. By comparing Geo-Nav w/o MSF with Res18-Baseline w/ SDE, we observe that the cross-modality attention module leads to major improvements on SPL. Meanwhile, Geo-Nav achieves significant performance gain by coupling MSF and CA modules. This indicates that the whole fusion mechanism plays a critical role in incorporating different modalities. By replacing CA with a single-scale vanilla self-attention module, there is only a minor improvement over the baseline.

Impacts of pre-trained model. As seen in Table 2, compared with Geo-Nav ‡, Geo-Nav achieves better performance. Geo-Nav ‡ utilizes ImageNet pre-trained weights for feature extractor. Hence, the improvements mainly come from more suitable visual representations extracted by SDE.

Impacts of source of scene geometry. As seen in Table 2, Geo-Nav $\dot{+}$ utilizes ground-truth depth as inputs in the depth branch. Geo-Nav $\dot{\wr}$ adopts the supervised depth prediction in the same way. Compared with Geo-Nav, Geo-Nav $\dot{+}$ only achieves slight improvements, and Geo-Nav $\dot{\wr}$ is slightly inferior to Geo-Nav. The results indicate that once depth quality is acceptable for navigation tasks, depth does not affect the final navigation severely.

Impacts of asymmetric design of attention. In Table 4, we exchange the roles of RGB and depth branches in generating key, value and query, denoted by Geo-Nav†. Specifically, the key and value are generated from depth embeddings, while the query is generated from RGB embeddings. In this case, Geo-Nav† attains inferior performance compared with Geo-Nav. Therefore, the choice of the asymmetric attention structure affects the feature representation.

RL-based Self-Supervised Learning

Some works (Ye et al. 2020; Pathak et al. 2017) employ Reinforcement Learning based Self-Supervised Learning (RL-SSL) to improve navigation efficiency, while Geo-Nav focuses on exploring scene geometry using vision-based self-supervised learning for navigation. For example, given the previous and current observations, s_{t-1} and s_t, Inverse dynamics model (IDM) (Ye et al. 2020) firstly obtains the corresponding feature embeddings $\phi(s_{t-1})$ and $\phi(s_t)$ from the feature extractor. Then, IDM predicts an action $\hat{a}_t$ from the features $\phi(s_{t-1})$ and $\phi(s_t)$ via a MLP layer and compute the cross entropy between $\hat{a}_t$ and the current action a_t to obtain expressive representations. Compared with Geo-Nav, Geo-Nav with IDM achieves better performance with the SPL of 85.7% and Success of 95%. Therefore, we observe that incorporating RL-SSL methods lead to better efficiency.

Conclusion

In this paper, we propose a scene-geometry driven point-goal navigation framework, dubbed Geo-Nav. Thanks to the scale-consistent self-supervised depth estimation network, Geo-Nav achieves superior visual perception ability. To the best of our knowledge, Geo-Nav is the first attempt to explore scene geometry from historical images of a monocular RGB camera in navigation tasks. Furthermore, the Cross-Modality Pyramid Fusion module (CPF) integrates the RGB images and depth information in a patch-wise fashion, and thus it can achieve more expressive visual representations. Extensive experiments demonstrate that Geo-Nav outperforms the state-of-the-art.

[2]Both are trained on V100

References

Anderson, P.; Chang, A.; Chaplot, D. S.; Dosovitskiy, A.; Gupta, S.; Koltun, V.; Kosecka, J.; Malik, J.; Mottaghi, R.; Savva, M.; et al. 2018. On evaluation of embodied navigation agents. *arXiv:1807.06757*.

Bhat, S. F.; Alhashim, I.; and Wonka, P. 2020. AdaBins: Depth Estimation using Adaptive Bins. *arXiv preprint arXiv:2011.14141*.

Bian, J.; Li, Z.; Wang, N.; Zhan, H.; Shen, C.; Cheng, M.-M.; and Reid, I. 2019. Unsupervised scale-consistent depth and ego-motion learning from monocular video. In *NeurIPS*.

Bian, J.-W.; Zhan, H.; Wang, N.; Chin, T.-J.; Shen, C.; and Reid, I. 2020. Unsupervised Depth Learning in Challenging Indoor Video: Weak Rectification to Rescue. *arXiv preprint arXiv:2006.02708*.

Bruhn, A.; Weickert, J.; and Schnörr, C. 2005. Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. *IJCV*.

Cadena, C.; Carlone, L.; Carrillo, H.; Latif, Y.; Scaramuzza, D.; Neira, J.; Reid, I.; and Leonard, J. J. 2016. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *T-RO*.

Chang, A.; Dai, A.; Funkhouser, T.; Halber, M.; Niessner, M.; Savva, M.; Song, S.; Zeng, A.; and Zhang, Y. 2017. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*.

Chaplot, D. S.; Gandhi, D.; Gupta, S.; Gupta, A.; and Salakhutdinov, R. 2020a. Learning to explore using active neural slam. In *ICLR*.

Chaplot, D. S.; Salakhutdinov, R.; Gupta, A.; and Gupta, S. 2020b. Neural Topological SLAM for Visual Navigation. In *CVPR*.

Chen, K.; de Vicente, J. P.; Sepulveda, G.; Xia, F.; Soto, A.; Vázquez, M.; and Savarese, S. 2019. A behavioral approach to visual navigation with graph localization networks. In *RSS*.

Davison, A. J.; Reid, I. D.; Molton, N. D.; and Stasse, O. 2007. MonoSLAM: Real-time single camera SLAM. *TPAMI*.

Du, H.; Yu, X.; and Zheng, L. 2020. Learning object relation graph and tentative policy for visual navigation. In *ECCV*.

Du, H.; Yu, X.; and Zheng, L. 2021. VTNet: Visual transformer network for object goal navigation. In *ICLR*.

Durrant-Whyte, H.; and Bailey, T. 2006. Simultaneous localization and mapping: part I. *RAM*.

Garcia-Fidalgo, E.; and Ortiz, A. 2015. Vision-based topological mapping and localization methods: A survey. *Robot. Auton. Syst.*

Godard, C.; Mac Aodha, O.; and Brostow, G. J. 2017. Unsupervised monocular depth estimation with left-right consistency. In *ICCV*.

Godard, C.; Mac Aodha, O.; Firman, M.; and Brostow, G. J. 2019. Digging into self-supervised monocular depth estimation. In *ICCV*.

Gordon, D.; Kadian, A.; Parikh, D.; Hoffman, J.; and Batra, D. 2019. SplitNet: Sim2Sim and Task2Task transfer for embodied visual navigation. In *ICCV*.

Gupta, S.; Davidson, J.; Levine, S.; Sukthankar, R.; and Malik, J. 2017. Cognitive mapping and planning for visual navigation. In *CVPR*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Jaderberg, M.; Simonyan, K.; Zisserman, A.; et al. 2015. Spatial transformer networks. In *NIPS*.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Klein, G.; and Murray, D. 2007. Parallel tracking and mapping for small AR workspaces. In *ISMAR*. IEEE.

Li, G.; Kang, G.; Liu, W.; Wei, Y.; and Yang, Y. 2020. Content-consistent matching for domain adaptive semantic segmentation. In *ECCV*.

Li, G.; Kang, G.; Zhu, Y.; Wei, Y.; and Yang, Y. 2021. Domain consensus clustering for universal domain adaptation. In *CVPR*.

Mur-Artal, R.; Montiel, J. M. M.; and Tardos, J. D. 2015. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Trans. Robot.*

Pathak, D.; Agrawal, P.; Efros, A. A.; and Darrell, T. 2017. Curiosity-driven exploration by self-supervised prediction. In *ICML*.

Rother, C.; Kolmogorov, V.; and Blake, A. 2004. " GrabCut" interactive foreground extraction using iterated graph cuts. *ACM TOG*.

Savva, M.; Kadian, A.; Maksymets, O.; Zhao, Y.; Wijmans, E.; Jain, B.; Straub, J.; Liu, J.; Koltun, V.; Malik, J.; et al. 2019. Habitat: A platform for embodied ai research. In *ICCV*.

Sax, A.; Emi, B.; Zamir, A. R.; Guibas, L.; Savarese, S.; and Malik, J. 2018. Mid-level visual representations improve generalization and sample efficiency for learning visuomotor policies. *arXiv preprint arXiv:1812.11971*.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv:1707.06347*.

Shen, W. B.; Xu, D.; Zhu, Y.; Guibas, L. J.; Fei-Fei, L.; and Savarese, S. 2019. Situational fusion of visual representation for visual navigation. In *ICCV*.

Shi, Y.; Li, H.; and Yu, X. 2021. Self-Supervised Visibility Learning for Novel View Synthesis. In *CVPR*.

Tang, T.; Yu, X.; Dong, X.; and Yang, Y. 2021. AutoNavigator: Decoupled Neural Architecture Search for Visual Navigation. In *WACV*.

Tian, Y.; Yu, X.; Fan, B.; Wu, F.; Heijnen, H.; and Balntas, V. 2019. Sosnet: Second order similarity regularization for local descriptor learning. In *CVPR*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. In *NeurIPS*.

Wagstaff, B.; and Kelly, J. 2021. Self-supervised scale recovery for monocular depth and egomotion estimation. In *IROS*.

Wijmans, E.; Kadian, A.; Morcos, A.; Lee, S.; Essa, I.; Parikh, D.; Savva, M.; and Batra, D. 2020. DD-PPO: Learning Near-Perfect PointGoal Navigators from 2.5 Billion Frames. In *ICLR*.

Xia, F.; Zamir, A. R.; He, Z.; Sax, A.; Malik, J.; and Savarese, S. 2018. Gibson env: Real-world perception for embodied agents. In *CVPR*.

Ye, J.; Batra, D.; Wijmans, E.; and Das, A. 2020. Auxiliary Tasks Speed Up Learning PointGoal Navigation. In *CoRL*.

Yu, X.; Tian, Y.; Porikli, F.; Hartley, R.; Li, H.; Heijnen, H.; and Balntas, V. 2019. Unsupervised extraction of local image descriptors via relative distance ranking loss. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*.

Yu, X.; Zhuang, Z.; Koniusz, P.; and Li, H. 2020. 6DoF Object Pose Estimation via Differentiable Proxy Voting Regularizer. In *BMVC*.

Zhan, H.; Garg, R.; Saroj Weerasekera, C.; Li, K.; Agarwal, H.; and Reid, I. 2018. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *CVPR*.

Zhou, T.; Brown, M.; Snavely, N.; and Lowe, D. G. 2017. Unsupervised learning of depth and ego-motion from video. In *CVPR*.

Zhu, F.; Zhu, Y.; Chang, X.; and Liang, X. 2020. Vision-language navigation with self-supervised auxiliary reasoning tasks. In *CVPR*.

Zhu, Y.; Mottaghi, R.; Kolve, E.; Lim, J. J.; Gupta, A.; Fei-Fei, L.; and Farhadi, A. 2017. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *ICRA*.