Imagine Before Go: Self-Supervised Generative Map for Object Goal Navigation

Supplementary Material

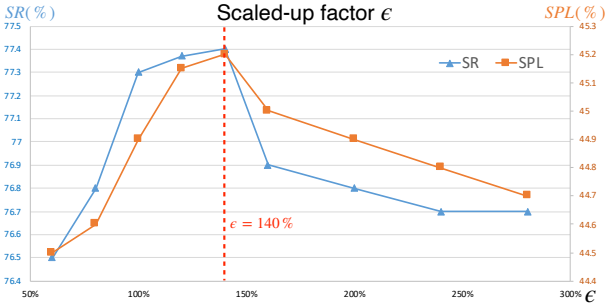

Figure 1. The impact of the scale-up factor ϵ on navigation performance on Gibson (val). To ensure that the results are solely influenced by ϵ, all patches are selected without being filtered by the sample strategy.

1. Hyper-parameter Analysis

Our SGM includes two hyper-parameters: the scale-up factor ϵ and the number of selected patches n. In this section, we conduct experiments to determine the optimal values for these hyper-parameters.

As shown in Fig. 3 of the main text, the scale-up factor ϵ specifies that the side length of cropped sub-map is ϵ times that of the smallest fitting square box around the known regions in the m_t (semantic local map). The SGM generates the unknown regions of the cropped sub-map to expand the agent's field of view. We set the value of ϵ from small to large and measure its impact on navigation performance on the SR and SPL metrics, as illustrated in Fig. 1. The experimental results indicate that the performance of navigation initially increases and then decreases with the increasing value of ϵ, reaching an optimal performance at $\epsilon = 140\%$. We infer that within a certain range, the predictions of SGM are reliable. However, when the expanded region is too large, patches far from known regions are challenging to precisely predict without the known adjacent patches. Therefore, performance declines when ϵ is excessively large. Conversely, when ϵ is too small, the SGM is insufficient to leverage its advantages, resulting in suboptimal navigation performance. Therefore, we set $\epsilon = 140\%$ based on these results. Note that considering that the regions belonging smallest fitting square box also contain unknown regions, when $\epsilon = 140\%$, SGM expands the agent's field of view to nearly twice its original size

The number of selected patches n determines how many patches are selected by the sampling strategy $multinomial(n, P)$. Then the selected patches are fed into the SGM to predict unknown patches. More patches mean that the SGM can access more information. Consequently, the navigation performance gradually increases with the number of patches, as shown in Fig.2 (a). However, as depicted in Fig. 2 (b), the computational complexity of the SGM also increases simultaneously. Therefore, to balance performance and complexity, we finally choose $n = 59$, which is approximately 30% of all patches.

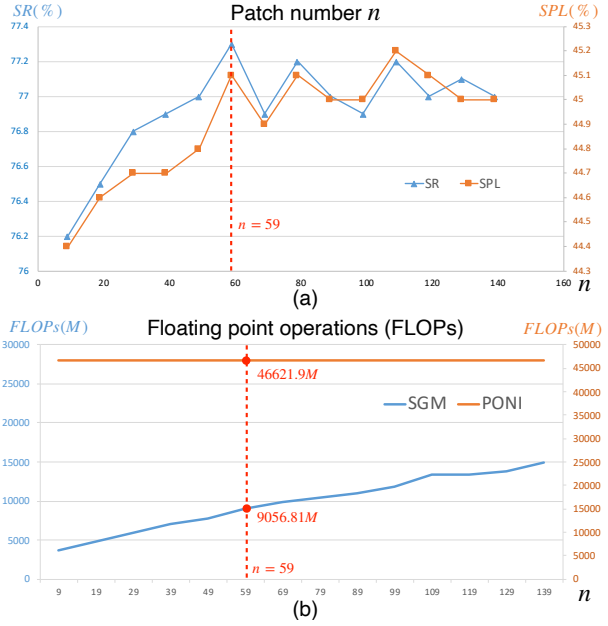

Figure 2. The impact of patch number n. (a) represents the impact on navigation performance in Gibson (val), where the scaled-up factor $\epsilon = 140\%$. (b) illustrates the effect of the patch number n on computational complexity, measured in terms of FLOPs (floating point operations).

2. Computation Complexity

Related modular works [6, 10, 11] typically employ an encoder-decoder model to directly use the entire local map as input for predicting the nearest frontier to the target [6], the minimal distance to the target [11], or the absolute coordinates of the target [10]. In contrast, our approach employs a sample strategy to select only a small portion (30%) of patches for input.

We compare the computational complexity of our SGM with PONI [6], where PONI proposes potential functions that utilize the local semantic map as input to predict potential frontiers (area potential function) and potential object locations (object potential function), which are implemented with UNet [7]. We employ the FLOPs (Floating Point Operations) to measure computational complexity,

Table 1. Comparison of response examples from GPT-4 [5] and ChatGLM [4]. We annotate query (observed) objects in blue and contextual objects in yellow in each sentence.

Prompt	GPT-4	ChatGLM
In an indoor scene, what objects are typically found surrounding a bed?(Answer in one sentence)	Around a bed, you often find a nightstand, lamp, and possibly a small chair or potted plant, along with items like a clock, book, or decorative vase.	A bed is likely to have a pillow, sheet, and blanket surrounding it.
In an indoor scene, what objects are typically found surrounding a toilet?(Answer in one sentence)	In an indoor scene, surrounding a toilet, you typically find items like a sink, a small trash can, and often a toilet paper holder or a towel rack.	A toilet is likely to have a toilet paper roll, a toilet seat cover, a bidet, and a sink.
In an indoor scene, what objects are typically found surrounding a couch, chair and oven, respectively?(Answer in one sentence)	Surrounding a couch, you might typically find a TV, coffee table, and potted plant; near a chair, there could be a book, small table, and vase; and around an oven, common objects include a refrigerator, sink, and kitchen cabinets.	A couch is likely to have a pillow, a sheet, and a blanket, a chair is likely to have a seat cushion, a table, and a lamp, and an oven is likely to have a burner, a cooking pot, and a counter.
In an indoor scene, what objects are typically found surrounding a cabinet, chest of drawers, sink and towel, respectively?(Answer in one sentence)	A cabinet is often surrounded by items like a chair, table, and cushion; a chest of drawers may be accompanied by a bed, clothes, and pictures; a sink typically has a mirror, towel, and toiletries nearby; and a towel is usually found near a shower, bathtub, or next to a sink.	A cabinet is likely to have a door, drawers, and a knob, a chest of drawers is likely to have a drawers, a knob, and a label, a sink is likely to have a drain, a faucet, and a counter, and a towel is likely to have a edge, a absorbent surface, and a label.
In an indoor scene, what objects are typically found surrounding a bed, sofa and fireplace, respectively?(Answer in one sentence)	Around a bed, you'll often find nightstands, a dresser or chest of drawers, and sometimes a seating area or plant; near a sofa, common items include a coffee table, cushions, and a TV monitor; and surrounding a fireplace, one might see comfortable seating, a rug, and decorative pictures or a mirror.	A bed is likely to have a pillow, sheet, and blanket, a sofa is likely to have a cushions, a sheet, and a blanket, and a fireplace is likely to have a mantle, a grate, a firebox, and a timer.

where a higher value of FLOPs indicates greater computational complexity and more time required for model inference. The results in Fig. 2 (b) show that our SGM has significantly lower computational complexity than PONI. Although SGM, implemented with ViT, has larger parameters (135.83M) compared to PONI (4.37M), its processes only of a subset of patches (when $n = 59$, the input dimension is $b \times (N_o + N_s) \times 16 \times 16 \times 59$) as opposed to PONI, which uses the entire map as input (the dimension is $b \times (N_o + N_s) \times 480 \times 480$). Therefore, SGM has a lower computational complexity.

3. Prompt and Response of LLMs

The Tab. 1 presents some response examples from two LLMs, GPT-4 [5] and ChatGLM [4]. Since LLMs are sensitive to prompts, in addition to the prompts listed in Tab. 1, we pre-input some predefined prompts to ensure that the responses are more applicable to goal categories in ObjectNav (as shown in Tab. 2). An example of a pre-input prompt is: 'There is an indoor scene, all object categories include but are not limited to chair, table, picture, cabinet, cushion, sofa, bed, chest of drawers, plant, sink, toilet, stool, towel, tv monitor, shower, bathtub, counter, fireplace, gym equipment, seating, clothes.'.

Table 2. Selected goal categories in Gibson, MP3D and HM3D.

Datasets	Training	Test
Gibson [8]	chair, couch, potted plant, bed, toilet, dining-table, tv, oven, sink, refrigerator, book, clock, vase, cup, bottle	chair, couch, tv, bed, toilet, potted plant
MP3D [1]	chair, table, picture, cabinet, cushion, sofa, bed, chest of drawers, plant, sink, toilet, stool, towel, tv monitor, shower, bathtub, counter, fireplace, gym equipment, seating, clothes	
HM3D [9]	chair, sofa, plant, bed, toilet, tv monitor, fireplace, bathtub, mirror	chair, sofa, bed, plant, toilet, tv monitor

Comparing the responses of GPT-4 and ChatGLM in Tab. 1, the GPT-4 provides more comprehensive responses and predicts a greater number of contextual objects. However, as shown in Tab. 1 of the main text, the choice of LLMs has minimal impact on navigation performance. Therefore, we ultimately chose the open-source ChatGLM for providing general knowledge.

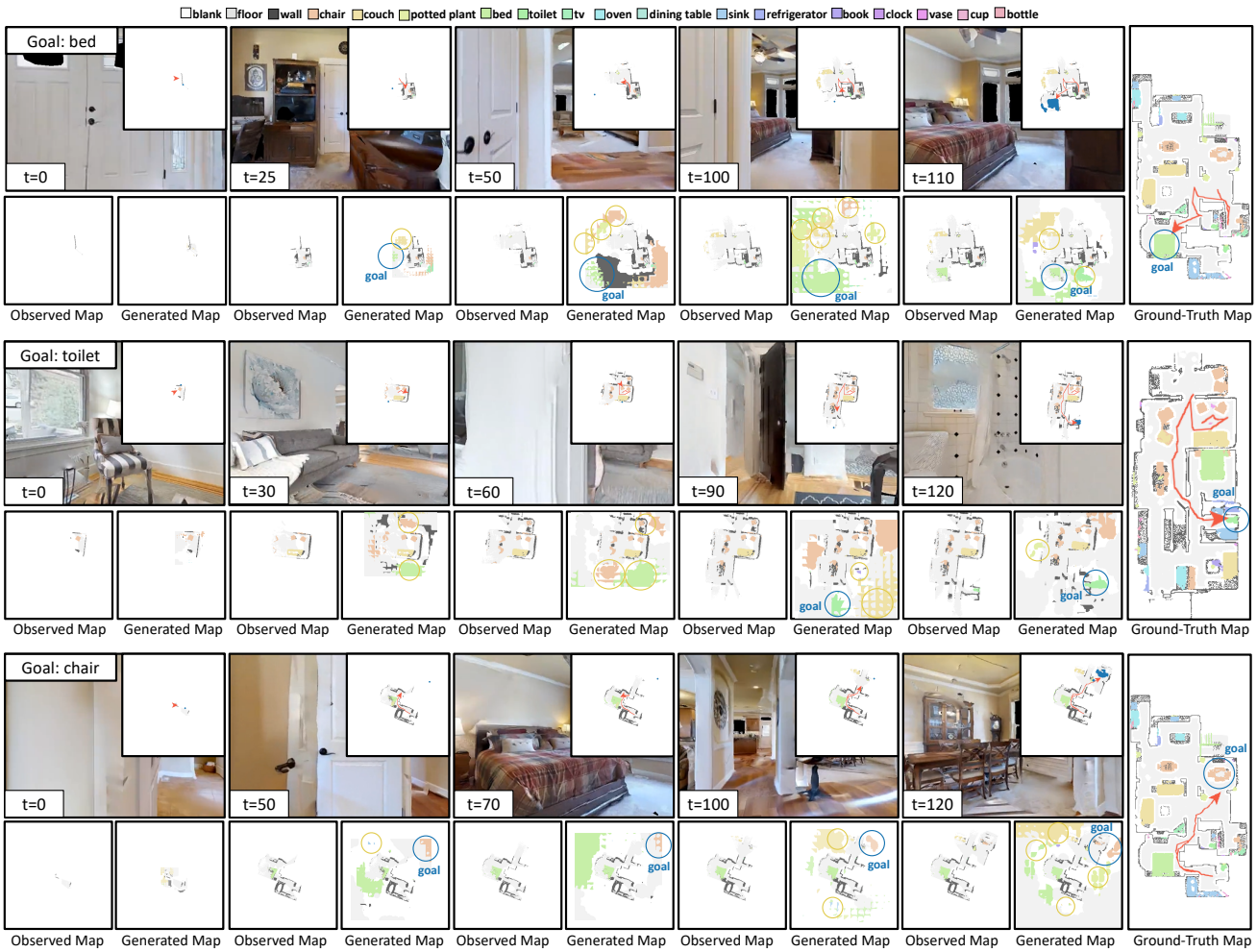

Figure 3. More visualizations. In each timestamp, we visualize the agent's RGB view, local semantic map (including trajectory and long-term goal (marked in blue dot)) and the generated map by SGM. The correctly predicted locations of target objects and context objects are marked in blue circles and yellow circles, respectively.

Furthermore, to reduce model complexity and accelerate inference speed, we pre-enumerate all possible object combinations and obtain corresponding responses from the LLMs in advance. We then extract and store the features of these responses, which allows the model to directly access the general knowledge (text features) based on the observed object categories.

4. Goal Categories

Our experimental setup follows previous works, and the adopted goal categories are detailed in Tab. 2. For Gibson, we follow [2, 6] and choose 15 object categories for training and 6 object categories for testing. For MP3D, we follow [6], where the same 21 object categories are used for both training and testing. For HM3D, we adopt the same experimental setup as [3], where 9 object categories are set for training and 6 object categories are used for testing.

5. More Visualizations

The Fig. 3 illustrates more visualizations of the generated maps of SGM during the navigation process. The range of the generated map increases as the observed local map expands. As shown in Fig. 3, SGM precisely predicts the orientation of the target object before it is observed. Notably, in the first and third rows of Fig. 3, SGM accurately predicts the location of the target (indicated by the blue circle) at the early stage of navigation. Based on this prediction, the agent avoids unnecessary detours and follows an almost optimal trajectory to reach the target, thus, our SGM significantly enhances the navigation efficiency. Furthermore, SGM not only accurately predicts the location of the goal but also identifies the positions of other contextual objects (indicated by the yellow circle), demonstrating its effectiveness in contextual reasoning.

References

[1] Angel X. Chang, Angela Dai, Thomas A. Funkhouser, Maciej Halber, Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from RGB-D data in indoor environments. In *2017 International Conference on 3D Vision, 3DV 2017, Qingdao, China, October 10-12, 2017*, pages 667–676. IEEE Computer Society, 2017. 2

[2] Devendra Singh Chaplot, Dhiraj Gandhi, Abhinav Gupta, and Russ R. Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 3

[3] Shizhe Chen, Thomas Chabal, Ivan Laptev, and Cordelia Schmid. Object goal navigation with recursive implicit maps. *CoRR*, abs/2308.05602, 2023. 3

[4] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335, 2022. 2

[5] OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023. 2

[6] Santhosh Kumar Ramakrishnan, Devendra Singh Chaplot, Ziad Al-Halah, Jitendra Malik, and Kristen Grauman. PONI: potential functions for objectgoal navigation with interaction-free learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 18868–18878. IEEE, 2022. 1, 3

[7] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III*, pages 234–241. Springer, 2015. 1

[8] Fei Xia, Amir Roshan Zamir, Zhi-Yang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 9068–9079. IEEE Computer Society, 2018. 2

[9] Karmesh Yadav, Ram Ramrakhya, Santhosh Kumar Ramakrishnan, Théophile Gervet, John M. Turner, Aaron Gokaslan, Noah Maestre, Angel Xuan Chang, Dhruv Batra, Manolis Savva, Alexander William Clegg, and Devendra Singh Chaplot. Habitat-matterport 3d semantics dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 4927–4936. IEEE, 2023. 2

[10] Albert J. Zhai and Shenlong Wang. Peanut: Predicting and navigating to unseen targets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10926–10935, 2023. 1

[11] Minzhao Zhu, Binglei Zhao, and Tao Kong. Navigating to objects in unseen environments by distance prediction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2022, Kyoto, Japan, October 23-27, 2022*, pages 10571–10578. IEEE, 2022. 1