SSCNav: Confidence-Aware Semantic Scene Completion for Visual Semantic Navigation

Yiqing Liang

Submitted in partial fulfillment of the
requirements for the degree of
Master of Computer Science
Thesis Track

COLUMBIA UNIVERSITY

2021

Abstract

SSCNav: Confidence-Aware Semantic Scene Completion for Visual Semantic Navigation

Yiqing Liang

This thesis focuses on visual semantic navigation, the task of producing actions for an active agent to navigate to a specified target object category in an unknown environment. To complete this task, the algorithm should simultaneously locate and navigate to an instance of the category. In comparison to the traditional point goal navigation, this task requires the agent to have a stronger contextual prior to indoor environments. This thesis introduces SSCNav, an algorithm that explicitly models scene priors using a confidence-aware semantic scene completion module to complete the scene and guide the agent's navigation planning. Given a partial observation of the environment, SSCNav first infers a complete scene representation with semantic labels for the unobserved scene together with a confidence map associated with its own prediction. Then, a policy network infers the action from the scene completion result and confidence map. The experiments demonstrate that the proposed scene completion module improves the efficiency of the downstream navigation policies. Code and data: `https://sscnav.cs.columbia.edu/`

Table of Contents

Acknowledgements

This work was supported in part by the Amazon Research Award, Columbia School of Engineering and National Science Foundation under CMMI-2037101.

I also want to thank the following people for their precious help in finalizing the thesis.

First of all, I would like to express my gratitude to Shuran Song, Carl Vondrick and Changxi Zheng for forming my thesis committee. I wish to extend my special thanks to Shuran Song for being my lighthouse and ally as my thesis advisor. Without you, I wouldn't have made it.

I also want to thank Boyuan Chen for the helping hand from the start to the end in all aspects, and Jimmy Wu for the valuable discussions that expedite the birth of this thesis.

Finally, I would not be where I am today without my family and my friends by my side. Thank you for all your love and support.

Chapter 1: Introduction and Background

1.1 Introduction

This thesis tackles the task of *visual semantic navigation*, specifically ObjectNav [1], where the goal is to navigate an agent (facing a random direction) from a random location in an unknown environment to a specified target object category (e.g., toilet) given first-person RGB-D image observations (shown in Fig. 1.1). In contrast to point goal navigation [2, 3, 4, 5], where the goal location is provided as a local coordinate and the agent's only job is to find a collision-free path leading to that coordinate, this task requires the agent to simultaneously answer two questions: (1) where to go (i.e., the target object location) and (2) how to get there (i.e., planning an efficient path to reach the target object).

To efficiently find the target object with an unknown appearance in an unseen environment, the system needs to leverage its functional priors of typical indoor environments to guide its high-level search policy (e.g., beds are often located in bedrooms) as well as spatial prior for low-level action planning (e.g., how to exit this room without collision). However, due to occlusion and limited camera field of view, the agent's observation of the environment contains only a small portion of the entire environment (e.g., a corner of a room in a big house). Thus this task requires a strong contextual prior of the complete 3D environment beyond an agent's partial observation.

This thesis introduces Confidence-Aware Semantic Scene Completion for Visual Semantic Navigation (**SSCNav**), an algorithm that explicitly models this scene prior using a confidence-aware semantic scene completion module and uses this scene representation to guide the agent's navigation planning. Given a partial view of an indoor scene in the form of RGB-D images, the semantic scene completion module predicts the semantic labels for the complete environment (in a top-down map) centered around the agent. By learning the statistics of many typical room lay-

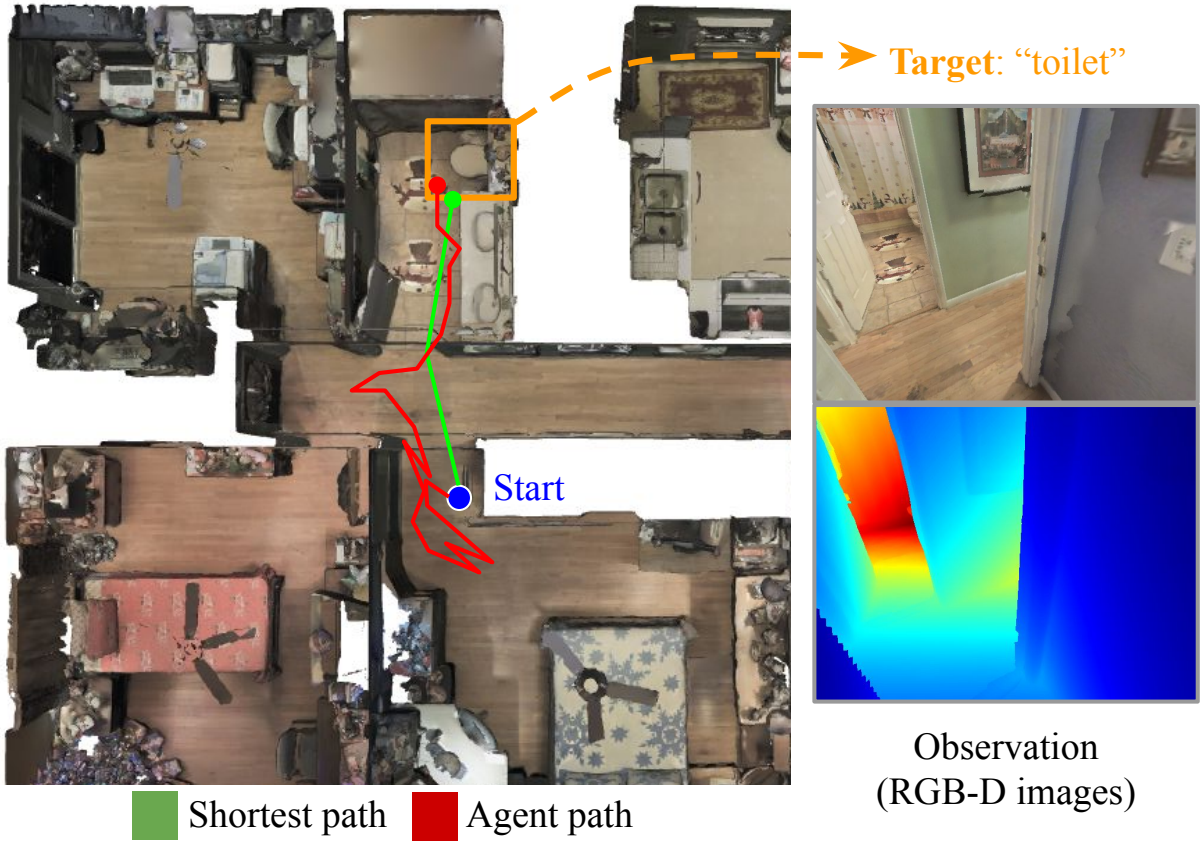

Figure 1.1: **Visual Semantic Navigation.** The goal of the algorithm is to navigate an agent equipped with a first-person RGB-D camera as well as pose sensors from a random location in an *unknown* scene to a given target object category (e.g., toilet).

outs, the module is able to leverage contextual clues to predict what is beyond the field of view for typical indoor environments. Then, the prediction of the complete 3D environment is used by a learning-based navigation algorithm for generating next-step actions. Since the inferred scene is used for planning, it is important for the completion module to output confidence in its predictions. To do so, the module produces a dense self-supervised confidence map together with the predicted environment map. The experiment included shows that the confidence estimation is critical for effectively guiding navigation and exploration by indicating to what degree the navigation policy could trust the inference result from the semantic scene completion module.

The navigation module is trained with deep reinforcement learning [6], where the policy network takes in the agent's current state and target object category as input and outputs the next action. The state is represented by a completed semantic and confidence map, and the action is represented by a dense spatial action map [7]. In this map, each pixel corresponds to an action that moves the agent towards that pixel's corresponding spatial location. In contrast to sparse action representation (e.g., steering command), this spatial action map representation is naturally aligned with the scene representation, and as shown in the ablation study, can significantly improve the performance [7, 8, 9, 10].

The primary contribution is SSCNav, a framework that applies semantic scene completion with confidence estimation for the task of object goal visual semantic navigation. By leveraging the contextual priors of the typical indoor environment, the algorithm is able to infer a scene representation beyond its partial observation. It's demonstrated that completed scene representation with confidence estimation enables more efficient planning policies for the downstream navigation task.

1.2 Background

Scene completion: SSCNav learns scene prior through scene completion, which is an idea explored in many recent works. Pathak et al. proposed ContextEnconder [11], which uses image completion to learn representations that capture contextual information. Song et al. proposed ex-

plicitly encoding 3D structure and semantic information in scene completion networks [12, 13]. However, these works do not study the effect of using these scene completion models in the context of motion planning. Specifically, they do not provide a way to measure the uncertainty of the network prediction, which is critical for downstream motion planning algorithms. Jayaraman and Grauman [14, 15] studied the use of panorama view completion for the task of selecting views. Ramakrishnan et al. [16] demonstrated the effectiveness of such learned policies in a few action-perception applications. However, the task of image completion is performed in RGB color space without modeling semantic or 3D structure. Moreover, the actions used in these works are constrained to camera rotation and hence they cannot directly support navigation tasks that require translation actions.

Point goal navigation: Point goal navigation is a well-studied problem with many prior works [2, 3, 4, 5]. In this task, taking in a target location and egocentric observations as input (images with color, depth, semantic segmentation, etc.), the agent needs to select one of the many possible actions and execute until it reaches the target coordinate. However, semantic goal navigation, specifically ObjectNav is much more challenging: since the target's location and appearance are both unknown to the agent, the system needs to estimate the target coordinate and the path towards it at the same time.

Semantic goal navigation: There have been rising interests in semantic navigation such as ObjectNav [17], where the goal is to navigate to a target object category instead of a global coordinate. For example, the self-adaptive visual navigation method (SAVN) [18] uses meta-learning to allow adaptation to unseen environments. Goal-Oriented Semantic Exploration (GOSE) [19] builds a global top-down map and uses it to improve exploration efficiency. Value Learning from Videos (VLV) [20] learns statistical regularities between object categories by watching YouTube videos. Another line of work utilizes semantic graphs to encode the scene structure for navigation [21, 22]. The target goal is specified by text and the scene prior is learned with a language graph module which mostly focuses on object functional priors captured in language. Different from these works, SSCNav explicitly captures the scene prior with a scene completion module, which natu-

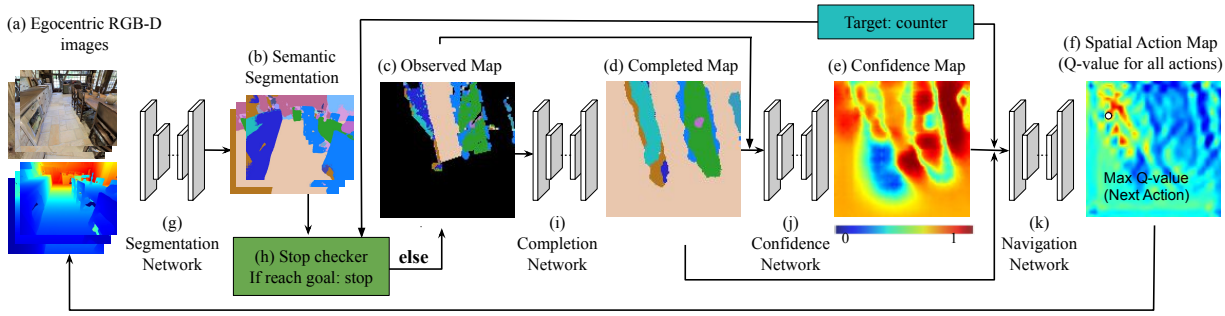

Figure 1.2: **SSCNav Overview.** At each step of an episode, the agent will get a new RGB-D image pair that is first used to predict the semantic segmentation (b) of the agent's current view. The past five observations are then aggregated and projected as a semantic top-down map (c), which is then completed by the scene completion network (i). The completed top-down map (d) and the observed semantic map (c) is then used by the confidence network (j) to estimate the confidence map (e) of the prediction. Next, the completed top-down map (d), the confidence map (e), and the target object category are used by the navigation network to predict a spatial action map (f), which is used to choose the next action. Finally, the stop checker (h) uses the new observation's semantic segmentation to determine whether the goal is reached.

rally encodes the knowledge of both functional priors and object spatial relationships that may not be reflected in text or 2D videos.

Chapter 2: Technical Approach

The approach consists of two major components: (1) a semantic scene completion module that takes in the agent's partial observation of the environment and infers a completed semantic representation $\hat{o}_t$ with confidence c_t, and (2) a navigation module that takes in the estimated $\hat{o}_t$, c_t, and the target object category to generate the next best action for the agent to move towards the goal. Fig. 1.2 shows the overview of the approach. Sec. 2.0.1 describes the confidence-aware semantic scene completion module and Sec. 2.0.2 presents the navigation module.

2.0.1 Confidence-Aware Semantic Scene Completion

Environment and agent setup: Habitat [17] with Matterport3D environments [23] is used to train and test the algorithm. In the setting, the agent uses a first-person-view RGB-D camera with an image resolution of 480×640. The camera is at a height of 1.25m from the floor, facing slightly downward at a $-30°$ angle. At each step, the agent receives a new RGB-D image. For each image, a semantic segmentation is obtained using off-the-shelf ACNet [24]. The ACNet is trained with 209,200 RGB-D images of 40 object categories from Matterprot3D training houses.

Agent-centric semantic top-down map: While the agent is moving in the environment, it aggregates the RGB-D observations, segmentation maps, and corresponding camera poses from the last 5 steps. The camera poses are used to combine past observations into one 3D point cloud. Here the groundtruth camera poses are provided by the environment, following the setting of Habitat 2020 ObjectNav challenge [1], as sensor pose estimation is out of scope for the task. After removing points that are higher than 1.55m from the agent's standing floor or lower than the floor, the remaining point cloud is projected to an agent-centric semantic top-down map o_t, at step t. This top-down map represents a 6m×6m local region centered around the agent and aligned with its ori-

entation (Fig. 2.1 c). Each pixel on the top-down map is labeled with the corresponding semantic category, represented with a one-hot vector with $(N+1)$ channels where N is the number of object categories, and the extra channel indicates the unknown category. This agent-centric semantic o_t is input to the scene completion network.

Scene completion network: The goal of the scene completion network is to take in the partial observation o_t as input and output a completed top-down map $\hat{o}_t$ where the unobserved regions are filled with predicted semantic information. The completion module is implemented with a fully convolutional neural network with 1 maxpooling layer, 4 down-sample residual blocks, and 5 up-sample residual blocks [25]. The network is trained to minimize the pixel-wise Cross-Entropy loss between the predicted semantic label and the groundtruth on the "unobserved region".

Self-calibrated confidence estimation: The semantic scene completion module estimates the agent's surrounding environment based on typical distributions of indoor environments learned from the training data. However, in practice, high variance in scene layouts and severe occlusion might lead to predictions with high uncertainty that may mislead the navigation module.

To mitigate this problem, the network's prediction uncertainty is explicitly estimated with another branch that outputs a corresponding confidence map $c_t \in [0,1]^{N\times M}$. The confidence map branch has the same structure as the completion network, and is trained with self-supervision. By comparing the network prediction with the groundtruth, a target correctness map ${c_t}^g \in [0,1]^{N\times M}$is obtained, where each value denotes whether the prediction is correct or not. Then, the confidence network is tasked with predicting this pixel-wise correctness value based on the input observation o_t and completed semantic map $\hat{o}_t$. The confidence network is trained to minimize the pixel-wise MSE loss between c_t and ${c_t}^g$ on the "unobserved region" only.

Training: To obtain training data for the scene completion network, the agent will randomly explore an environment, aggregate its observations and estimations (e.g., color, depth), and reconstruct the 3D scene. During exploration, the scene completion network is trained by randomly removing observed views. The network is tasked to infer the missing data. The unexplored region of the environment will be ignored for this training step (as shown in Fig. 2.1). This learning

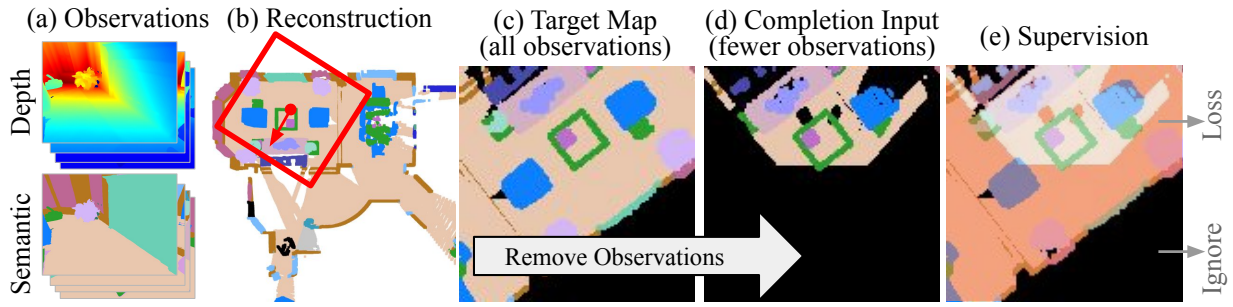

Figure 2.1: **Scene Completion Data.** The agent-centric depth and corresponding semantic observations are collected to obtain a global reconstruction of the 3D environment. Then the target semantic top-down map is generated with all the observations, and the input top-down map to the completion network is generated with fewer observation images (one to four views). To train the completion network, losses are only computed in regions that are observed in the target top-down map and ignore the unobserved regions.

scheme allows the system to learn from any new environment through navigation without fully reconstructed environments. In the experiment, 52,300 groundtruth top-down maps are generated for training, and each groundtruth map is paired with four different input observations with randomly removed views.

2.0.2 Visual Semantic Navigation with Reinforcement Learning

DQN formulation: The navigation task is formulated as a Markov Decision Process. Given a target object category g and current state s_t, the navigation policy $\pi(s_t)$ outputs the agent's next action a_t, then receives the next state s_{t+1} and a scalar reward r_t from the environment. The learning objective is to find a sequence of actions that maximize the total expected future rewards $\Sigma_{i=t}^{\infty}\gamma^{i-t}r_i$, where $\gamma \in (0, 1)$, a discounted sum over all future returns rewards.

Off-policy Q-learning [26] is used to learn a policy $\pi(s_t)$ that chooses actions which maximize a Q-function ${}_{a_t}Q_\theta(s_t, a_t)$ (i.e. state-action value function), which is a fully convolutional neural network [25, 27] parameterized by θ. The agents are trainedusing the double DQN learning objective [28]. Formally, at each step t, the objective is to minimize: $\mathcal{L} = |r_t+\gamma Q_{\theta^-}(s_{t+1},{}_{a_{t+1}}Q_\theta(s_{t+1}, a_{t+1}))-Q_\theta(s_t, a_t)|$, where (s_t, a_t, r_t, s_{t+1}) is a transition uniformly sampled from the agent's replay buffer. Target network parameters θ^- are held fixed between individual updates and updated less fre-

quently.

State representation: The state representation s_t includes three parts: (1) the completed semantic top-down map $\hat{o}_t$, (2) the corresponding confidence estimation c_t, and (3) the target object category g. The target object category is encoded as a one-hot vector with $(N + 1)$ channels (N object categories plus background). This vector is tiled to have the same shape as c_t and $\hat{o}_t$. All three elements are concatenated along the channel as the state representation.

Action representation: The action space $\mathcal{A}$ is represented as a spatial action map similar to Wu et al. [7]. The spatial action map representation shares the same spatial size as the state, where each pixel in the top-down map corresponds to a local navigational endpoint in the agent-centric map. At each step, after the agent selects a pixel location from the action map, it turns toward the corresponding point in the environment and moves forward for one step with a maximum step size of 0.25m (collision is possible). To select the best action, the navigation network estimates the expected Q value for all the possible actions represented as a heat map (e.g., Fig. 3.3). At test time, the policy picks actions by greedily choosing the pixel with the highest Q-value.

This Q-value representation is spatially aligned and anchored to the scene representation, enabling significantly faster learning of complex behaviors in navigation, which is demonstrated in ablation study. Additionally, the action representation allows the Q-value of each local navigation endpoint and, thus, the behavior of the navigation policy, to be visualized on a heat map.

Reward: At time step t, the agent gets a reward r_t that is the sum of:

- A life penalty of -0.01 to encourage a shorter path.
- A penalty of -0.25 if the distance of the step is < 0.125m, to encourage movement and punish collision.
- A delta-distance reward proportional to the change in the agent's distance to the closest target object.
- A success reward of +10 if the agent reaches success.

Stop checker: As part of the task, the system needs to decide whether it has successfully reached its goal. To model this ability, a Stop Checker (Fig. 1.2 h) that uses semantic and depth information to check whether the target object category is reached is designed. From the agent's current standpoint, its camera is rotated along the z-axis 4 times and get pairs of observations. The Stop Checker returns true and terminates an episode if there are enough ($> 5,000$) pixels belonging to the target object category within 1m distance to the agent in at least 1 pair of observation. The episode will also be terminated if the agent has navigated for 500 steps.

Task completion: At each action step, the environment will check whether the agent has succeeded and provide rewards accordingly. The following success metric is used according to [1](note that validity is trivially met in the setting): (1) Intentionality: the agent's Stop Checker has returned true. (2) Proximity: the agent is within 1 meter away from one target instance (3) Visibility: at least one target instance is visible.

Training: Each training/testing episode $\mathcal{T}$ contains 4 elements: a start location p, a start direction f, a scene h, and a target object category g. For a training episode $\mathcal{T}_k$, the target object categories are uniformly sampled. A training house is then sampled from Matterport3D's training set and randomly choose a legal start point p. A start point is considered legal if it (1) is navigable (2) is in one of the valid room types (bathroom, bedroom, dining room, kitchen, living room, laundry room, and family room) to avoid starting outdoor, and (3) is far from all target objects (both the Euclidean and Geodesic distance to any target instance are greater than $r + \mathrm{D}_{\mathrm{succ}}$, where r is the instance's radius and $\mathrm{D}_{\mathrm{succ}}$ is success distance (1m)). If such a point is not found after 100 trials, the search is repeated. Otherwise, the agent is turned to face a random direction and ready to start.

Chapter 3: Results

In this section, experimental results are provided to validate the proposed approach. All experiments are run on the Habitat platform [17] using the standard train-test split of Matterport3D [23]. In all the experiments, the agent is tested in *novel houses*. Sec. 3.0.1 summarize the result of semantic scene completion and Sec. 3.0.2 study the effect of using scene completion in navigation.

3.0.1 Semantic Scene Completion

Metric: The performance of scene completion is evaluated with 715 samples from unseen houses in Matterport3D. The category-level pixel intersection over union (IoU) is choosed as the evaluation metric. The IoU is computed in the regions that are *unobserved* region in the input top-down map but observed in the groundtruth [13].

Result: Fig. 3.2 summarizes the quantitative results of the scene completion network by using estimated semantic segmentation (completion-seg) and groundtruth semantic segmentation (completion-gt) as input observation. Fig. 3.1 shows the qualitative results and the estimated confidence map. The results show that the scene completion network is able to: (1) complete partially observed object geometry – Fig. 3.1 row 4. (2) infer unobserved object based on contextual information – Fig. 3.1 row 3 shows an example where the algorithm is able to infer the existence of an unobserved nightstand beside the bed based on the observed nightstand and infer the unobserved table and chair in row 1,2. (3) correct the segmentation error from the semantic segmentation network in the input observation – Fig. 3.1 row 2 shows an example where the completion network corrects the incorrect prediction of misc to correct table class. Meanwhile, the completion model also produces incorrect prediction (labeled as red in Fig. 3.1). The confidence score generated by the network, in general, reflects the confidence of the prediction (lower score for less-confident).

Table 3.1: Navigation Result (Success Rate / SPL). /SA: using sparse action, -CF: without scene completion and confidence, -F: without confidence, /BC: binary confidence based on visibility. G+ using ground truth segmentation for observed scene.

Model	Bed	Counter	Shower	Sink	Sofa	Table	Toilet	Avg
G+SSCNav-CF (GOSE [19])	0.083/0.024	0.136/0.062	0.200/**0.101**	0.207/0.098	**0.377/0.182**	0.591/0.380	0.036/0.003	0.387/0.232
G+SSCNav-F	0.083/**0.061**	0.121/0.054	0.043/0.015	0.268/**0.158**	0.208/0.129	0.574/0.394	0.071/0.004	0.357/0.235
G+SSCNav	**0.104**/0.057	**0.227/0.092**	**0.257**/0.073	**0.280**/0.149	0.245/0.115	**0.656/0.425**	**0.143/0.061**	**0.438/0.259**
SAVN [18]	0.000/0.000	0.000/0.000	0.000/0.000	0.000/0.000	0.000/0.000	0.018/0.018	0.000/0.000	0.009/0.009
SSCNav/SA	0.000/0.000	0.015/0.002	0.000/0.000	0.049/0.019	0.000/0.000	0.206/0.086	0.000/0.000	0.109/0.045
SSCNav-CF (GOSE [19])	0.021/0.002	0.015/0.005	0.043/0.009	0.037/0.027	**0.151/0.055**	0.394/0.219	0.000/0.000	0.218/0.118
SSCNav-F	0.042/0.035	0.076/0.034	0.000/0.000	0.122/0.028	0.019/0.009	0.385/0.216	0.000/0.000	0.217/0.117
SSCNav/BC	0.000/0.000	0.121/0.027	0.057/0.006	0.232/0.083	0.038/0.009	**0.409/0.275**	0.036/0.010	0.252/0.150
SSCNav	**0.042/0.040**	**0.152/0.040**	**0.200/0.059**	**0.268/0.097**	0.057/0.029	0.388/0.261	**0.107/0.037**	**0.271/0.157**

In the next section, the effect of using this semantic scene completion module in navigation wll be studied.

3.0.2 Semantic Navigation Result

Metric: The models are evaluated with 687 navigation episodes from unseen houses in Matterport3D provided by Habitat, with categories in {Bed, Counter, Shower, Sink, Sofa, Table, Toilet}. The standard navigation success rate and SPL (Success weighted by normalized inverse Path Length) are used as metric [29]. Note that there might be multiple object instances that belong to the target category. In this case, the closest target object when the episode starts will be chosen to compute the shortest path.

Comparison with prior work: The approach is evaluted and compared to SAVN from Wortsman et al. [18]. For the comparison, SAVN is finetuned and tested using the same setting as SSCNav and their performance is reported in Tab. 3.1 ([SAVN] v.s. [SSCNav]). Although SAVN only has live penalty and success reward during training originally, extra continuous distance reward (the same as SSCNav's) is added to narrow the gap. The result shows that the proposed method [SSCNav] significantly outperforms [SAVN] in all object categories. It's conjectured that the reasons are two-fold. First, SAVN uses sparse action representation where $\mathcal{A}$ = { MoveAhead, RotateLeft,..., Done }, and the mapping from observations to action classes is much harder to learn compared to the dense prediction enabled by the spatial action maps used in the proposed method. Second, SAVN does not explicitly model semantic object classes in the scene representation, which might

make the learning more difficult.

Another prior work is GOSE [19], the winner of Habitat 2020 ObjectNav Challenge [1]. GOSE uses a similar top-down semantic map and spatial action representation. However, since the code is not publicly available, and the main difference is the lack of semantic scene completion module, one could refer to the [SSCNav-CF] performance (SSCNav without (-) scene (C)ompletion and con(F)idence) in Tab. 3.1, as the attempt of re-implementing their system.

Does confidence-aware scene completion help? To test the effect of using semantic scene completion module in navigation, the navigation performance of models trained with and without the semantic scene completion module is compared ([SSCNav-CF] v.s. [SSCNav-F] v.s. [SSCNav], C, scene (C)ompletion F, con(F)idence). By comparing [SSCNav-CF] and [SSCNav-F], it's observe that directly using the scene completion does not affect the navigation success rate. It's conjectured that the error and noises in the scene completion result can still be misleading for navigation planning, and the confidence information naturally carried in the scene completion output (after softmax) is not sufficient.

This problem is largely mitigated with the confidence estimation. [SSCNav] outperforms [SSCNav-CF] with +5.3% in success rate and +3.9% in SPL. Even when completion errors exist, [SSCNav] could recovers and outperforms [SSCNav-CF] in most categories in light of confidence awareness. For some categories that [SSCNav] is not outperforming [SSCNav-CF] the SPL numbers are much higher (e.g., table -0.6% success rate, +4.2% SPL). The result indicates that the confidence-aware semantic scene completion module is particularly helpful for objects with strong contextual bias (e.g., toilets only appear in restrooms). On the other hand, for common objects such as sofas and tables that can appear in almost any room, the contextual bias is less useful.

What does confidence estimation learn? Given that confidence estimation is critical to allow the model to benefit from scene completion, a question is whether the confidence estimation is better than naively masking seen area as 'confident' and unseen area as 'unconfident'? To answer this question, an additional baseline [SSCNav/BC] (BC, binary confidence) is included, where the confidence is modeled by scene visibility: seen region with value 1, unseen region with value 0. The

result shows that [SSCNav] outperforms [SSCNav/BC] +1.9% in success rate and +0.7% in SPL. This result validates that while the scene visibility is highly correlated with network uncertainty, the confidence estimation still carries more useful information.

Effect of spatial action representation. To test the benefit of using spatial action map as the action representation, [SSCNav] is compared with [SSCNav/SA] (/SA, spares action). [SSCNav/SA] is the replaces navigation model's output map with a sparse action probability set, a common strategy for such policy. The following action space is used: {Turn right for k degrees, then move forward $|k \in [0, 45, 90, 135, 180, 225, 270, 315]\}$. At each time step, the agent picks the action with maximum probability instead of the pixel with maximum Q-value. Along with losing the interpretability of the navigation model, a significant drop is witnessed both in success rate (-16.2%) and SPL (-14.8%). This result is consistent with prior work [7].

Performance oracle. Here the navigation performance is evaluted when the groundtruth segmentation for the observed region is available. Tab. 3.1, [G+SSCNav]s show the navigation performance. While groundtruth input clearly improves the algorithm performance, the result still follows a similar tendency: even with perfect semantic segmentation as input, semantic scene completion still improves navigation performance. This result indicates that with the advancement of general semantic segmentation algorithms, the system's performance will also be improved.

Conclusion or Epilogue

SSCNav is introduced for the task of visual semantic navigation. The algorithm leverages a confidence-aware semantic scene completion module to achieve a better understanding of the agent's surrounding environment and uses this scene completion result to facilitate its action planning in navigation. The experiments demonstrate that by learning the contextual priors of the typical indoor environment, SSCNav is able to improve the performance of visual semantic navigation in terms of both success rate and efficiency.

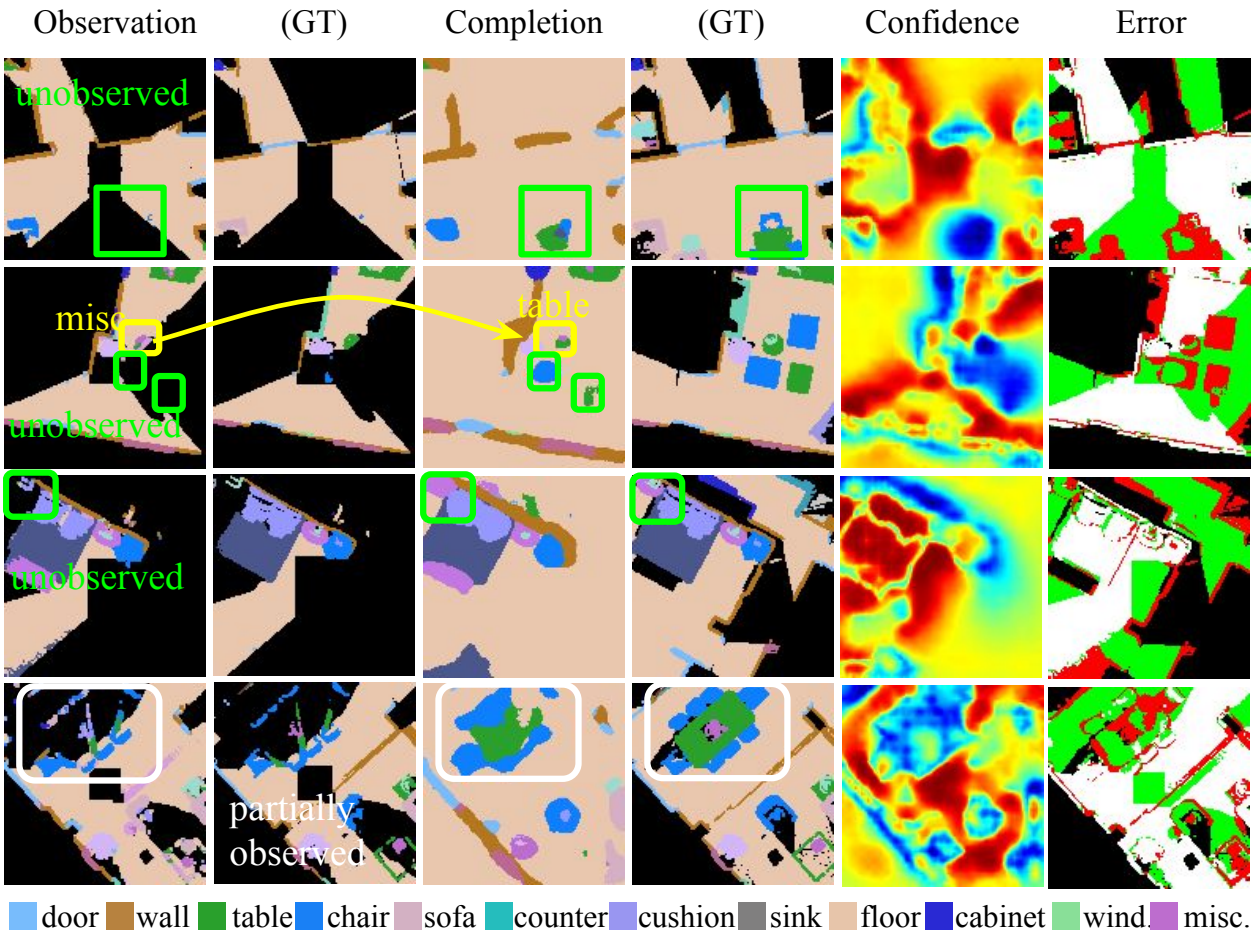

Figure 3.1: **Scene Completion Result.** From left to right are predicted and groundtruth segmentation of input observations, predicted and groundtruth scene completion, confidence estimation, and error map (white: observed area, red: incorrect completion, green: correct completion). The results show that the scene completion network is able to complete partially observed object (row 4), infer unobserved object (row 1,2,3), and correct segmentation error in the input observations (row 2).

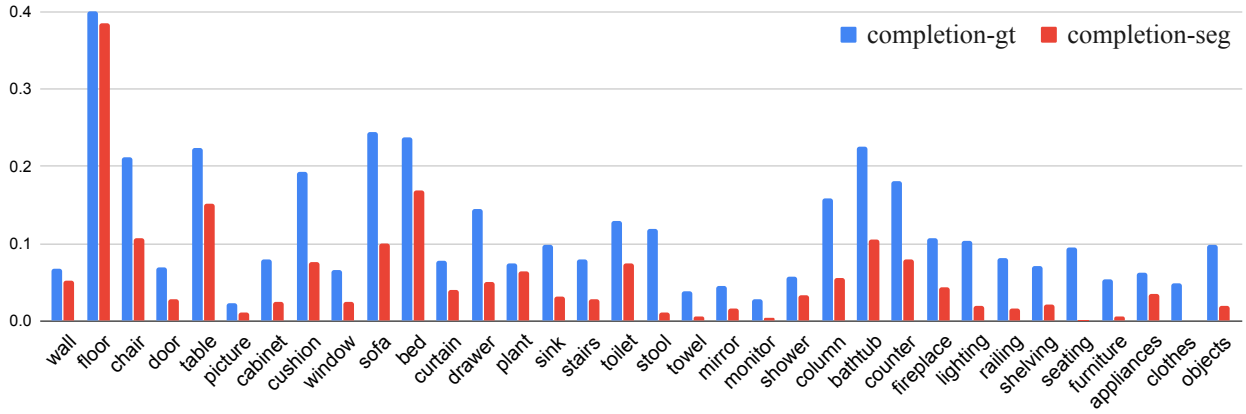

Figure 3.2: **Semantic Scene Completion Results.** The plot shows intersection over union (IoU) of estimated semantic labels and groundtruth labels in the unobserved regions when taking groundtruth segmentation as input (blue) and estimated segmentation as input (red). Six categories (beam, blinds, gym-equip, board, ceiling, misc) that have zero IoU are not included.

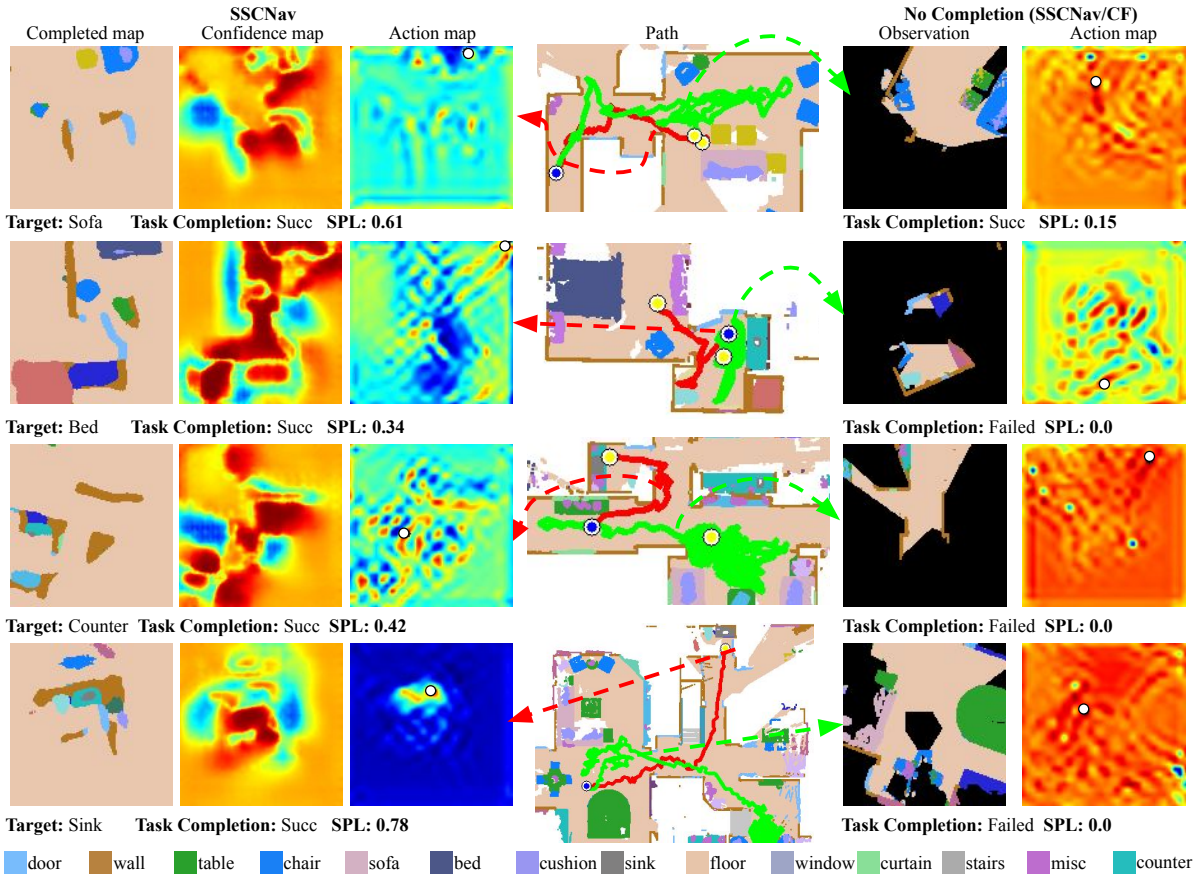

Figure 3.3: **Qualitative Result of Semantic Navigation.** In the path visualization (column 4), the red path shows the navigation path of the agent using the semantic scene completion module (SSCNav), and the green path is taken by the agent without semantic scene completion (SSCNav-CF). The blue dot indicates the start position and the yellow dot indicates the end position. Note that in the 1st row, agents from both approaches successfully reach their goals (a sofa) but SSCNav-CF makes a huge mistake and thus struggles to find the target.

References

[1] D. Batra, A. Gokaslan, A. Kembhavi, O. Maksymets, R. Mottaghi, M. Savva, A. Toshev, and E. Wijmans, "ObjectNav Revisited: On Evaluation of Embodied Agents Navigating to Objects," in arXiv:2006.13171, 2020.

[2] M. Savva, A. X. Chang, A. Dosovitskiy, T. Funkhouser, and V. Koltun, "Minos: Multimodal indoor simulator for navigation in complex environments," arXiv preprint arXiv:1712.03931, 2017.

[3] 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 Conference on Computer Vision and Pattern Recognition, 2018, pp. 9068–9079.

[4] C. Yan, D. Misra, A. Bennnett, A. Walsman, Y. Bisk, and Y. Artzi, "Chalet: Cornell house agent learning environment," arXiv preprint arXiv:1801.07357, 2018.

[5] E. Wijmans, A. Kadian, A. S. Morcos, S. Lee, I. Essa, D. Parikh, M. Savva, and D. Batra, "Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames," in ICLR 2019, 2019.

[6] S. Qi, Y. Zhu, S. Huang, C. Jiang, and S.-C. Zhu, "Human-centric indoor scene synthesis using stochastic grammar," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5899–5908.

[7] J. Wu, X. Sun, A. Zeng, S. Song, J. Lee, S. Rusinkiewicz, and T. Funkhouser, "Spatial action maps for mobile manipulation," in RSS, 2020.

[8] A. Zeng, S. Song, K.-T. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo, et al., "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," in 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2018, pp. 1–8.

[9] A. Zeng, S. Song, S. Welker, J. Lee, A. Rodriguez, and T. Funkhouser, "Learning synergies between pushing and grasping with self-supervised deep reinforcement learning," in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2018, pp. 4238–4245.

[10] A. Zeng, "Learning visual affordances for robotic manipulation," Ph.D. dissertation, Princeton University, 2019.

[11] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. Efros, "Context encoders: Feature learning by inpainting," 2016.

[12] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, "Semantic scene completion from a single depth image," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1746–1754.

[13] S. Song, A. Zeng, A. X. Chang, M. Savva, S. Savarese, and T. Funkhouser, "Im2pano3d: Extrapolating 360 structure and semantics beyond the field of view," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3847–3856.

[14] D. Jayaraman and K. Grauman, "Look-ahead before you leap: End-to-end active recognition by forecasting the effect of motion," in ECCV, 2016.

[15] ——, "Learning to look around: Intelligently exploring unseen environments for unknown tasks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[16] S. K. Ramakrishnan, D. Jayaraman, and K. Grauman, "Emergence of exploratory look-around behaviors through active observation completion," Science Robotics, vol. 4, no. 30, eaaw6326, 2019.

[17] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra, "Habitat: A Platform for Embodied AI Research," in ICCV, 2019.

[18] M. Wortsman, K. Ehsani, M. Rastegari, A. Farhadi, and R. Mottaghi, "Learning to learn how to learn: Self-adaptive visual navigation using meta-learning," in CVPR, 2019.

[19] D. S. Chaplot, D. Gandhi, A. Gupta, and R. Salakhutdinov, "Object goal navigation using goal-oriented semantic exploration," in In Neural Information Processing Systems, 2020.

[20] M. Chang, A. Gupta, and S. Gupta, "Semantic visual navigation by watching youtube videos," in Advances in Neural Information Processing Systems, 2020.

[21] W. Yang, X. Wang, A. Farhadi, A. Gupta, and R. Mottaghi, "Visual semantic navigation using scene priors," arXiv preprint arXiv:1810.06543, 2018.

[22] Y. Qiu, A. Pal, and H. I. Christensen, "Target driven visual navigation exploiting object relationships," ArXiv, vol. abs/2003.06749, 2020.

[23] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3D: Learning from RGB-D data in indoor environments," 2017.

[24] X. Hu, K. Yang, L. Fei, and K. Wang, "Acnet: Attention based network to exploit complementary features for rgbd semantic segmentation," in 2019 IEEE International Conference on Image Processing (ICIP), IEEE, 2019, pp. 1440–1444.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in CVPR, 2016.

[26] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al., "Human-level control through deep reinforcement learning," Nature, vol. 518, no. 7540, p. 529, 2015.

[27] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.

[28] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in Thirtieth AAAI conference on artificial intelligence, 2016.

[29] P. Anderson, A. X. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva, and A. R. Zamir, "On evaluation of embodied navigation agents," ArXiv, vol. abs/1807.06757, 2018.