




Learning from Unlabeled 3D Environments for Vision-and-Language Navigation

Shizhe Chen¹, Pierre-Louis Guhur¹, Makarand Tapaswi²,
Cordelia Schmid¹, and Ivan Laptev¹

¹ Inria, École normale supérieure, CNRS, PSL Research University
² IIIT Hyderabad

https://cshizhe.github.io/projects/hm3d_autovln.html

Abstract. In vision-and-language navigation (VLN), an embodied agent is required to navigate in realistic 3D environments following natural language instructions. One major bottleneck for existing VLN approaches is the lack of sufficient training data, resulting in unsatisfactory generalization to unseen environments. While VLN data is typically collected manually, such an approach is expensive and prevents scalability. In this work, we address the data scarcity issue by proposing to automatically create a large-scale VLN dataset from 900 unlabeled 3D buildings from HM3D [45]. We generate a navigation graph for each building and transfer object predictions from 2D to generate pseudo 3D object labels by cross-view consistency. We then fine-tune a pretrained language model using pseudo object labels as prompts to alleviate the cross-modal gap in instruction generation. Our resulting HM3D-AutoVLN dataset is an order of magnitude larger than existing VLN datasets in terms of navigation environments and instructions. We experimentally demonstrate that HM3D-AutoVLN significantly increases the generalization ability of resulting VLN models. On the SPL metric, our approach improves over state of the art by 7.1% and 8.1% on the unseen validation splits of REVERIE and SOON datasets respectively.

Keywords: Vision-and-Language, Navigation, 3D Environments

1 Introduction

Having a robot carry out various chores has been a common vision from science fiction. Such a long-term goal requires an embodied agent to understand our human language, navigate in the physical environment and interact with objects. As an initial step towards this goal, the vision-and-language navigation task (VLN) [3] has emerged and attracted growing research attention. Early VLN approaches [3, 28] provide agents with step-by-step navigation instructions to arrive at a target location, such as “*Walk out of the bedroom. Turn right and walk down the hallway. At the end of the hallway turn left. Walk in front of the couch and stop*”. While these detailed instructions reduce the difficulty of the task, they lower the practical value for people when commanding robots in real

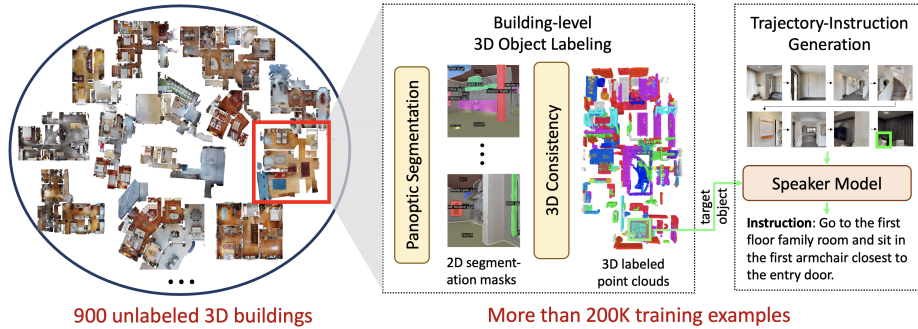


Fig. 1: **HM3D-AutoVLN dataset.** We use 900 unlabeled 3D buildings from the HM3D [45] dataset. We improve labels obtained with a 2D segmentation model based on 3D cross-view consistency (middle, see Fig. 2). Then we rely on these pseudo labels to generate instructions via a speaker model (right, see Fig. 3). We automatically create over 200K realistic training samples for VLN.

life. Thus, more recent VLN methods [43,57] focus on high-level instructions and request an agent to find a specific object at the target location, *e.g.*, “Go to the living room and bring me the white cushion on the sofa closest to the lamp”. The agent needs to explore 3D environments on its own to find the “cushion”.

Compared to step-by-step instructions, following such high-level goal-driven instructions is more challenging. As there is no detailed guidance, an agent needs to learn about the structure of the environment for effective exploration. However, most existing VLN tasks such as REVERIE [43] or SOON [57] are based on 3D scans from the Matterport3D (MP3D) dataset [4], and contain less than 60 buildings and approximately 10K training trajectories. The limited amount of training data makes VLN models overfit to seen environments, resulting in less generalizable navigation policies in unseen environments. Manually collecting more VLN data is however expensive and not scalable. To address this data scarcity issue, previous works have investigated various data augmentation methods, such as synthesizing more instructions and trajectories in seen environments by a speaker model [16], environment dropout [49] or editing [30], and mixing up seen environments [33]. Nevertheless, these approaches are still based on a small amount of 3D environments that cannot cover a wide range of objects and scenes. To address visual diversity, VLN-BERT [36] utilizes image-caption pairs from the web to improve generalization ability, while Airbert [18] shows that pairs from indoor environments, the BnB dataset, are more beneficial for VLN tasks. However, it is hard for image-caption pairs to mimic the real navigation experience of an agent in 3D environments making it challenging to learn a navigation policy with action prediction.

In this work, we propose a new data generation approach to improve the model’s generalization ability to unseen environments by learning from large-scale unlabeled 3D buildings (see Fig. 1). We take advantage of the recent HM3D

dataset [45] consisting of 900 buildings in 3D. However, this data comes without any labels. In order to generate high-quality instruction-trajectory pairs on a diverse set of unseen environments, we use large-scale pretrained vision and language models. We first use an image segmentation model [10] to detect 2D objects for images in the environment, and utilize cross-view consistency in 3D to increase the accuracy of pseudo 3D object annotations. Then, we fine-tune a language model, GPT-2 [44], with pseudo object labels as prompts to generate high-level navigation instructions to this object. In this way, we construct the HM3D-AutoVLN dataset that uses 900 3D buildings, and consists of 36,562 navigable nodes, 172,000 3D objects and 217,703 object-instruction-trajectory triplets for training – an order of magnitude larger than prior VLN datasets. We train multiple state-of-the-art VLN models [8,9,22,49] with the generated HM3D-AutoVLN data and show significant gains. Specifically, we improve over the state-of-the-art DUET model [9] on REVERIE and SOON datasets by 7.1% and 8.1% respectively. In summary, our contributions are as follows:

- We introduce an automatic approach to construct a large-scale VLN dataset, HM3D-AutoVLN, from unlabeled 3D buildings. We rely on 2D image models to obtain pseudo 3D object labels and on pretrained language models to generate instructions.
- We carry out extensive experiments on two challenging VLN tasks, REVERIE and SOON. The training on HM3D-AutoVLN dataset significantly improves the performance for multiple state-of-the-art VLN models.
- We provide insights on data collection and challenges inherent to leveraging unlabeled environments. It suggests that the diversity of environments is more important than the number of training samples alone.

2 Related Work

Vision-and-language navigation. The VLN tasks have recently been popularized with the emergence of various supportive datasets [3,6,23,27,28,39,48,50]. Different instructions define the variations of VLN tasks. Step-by-step instructions [3,6,28] require an agent to strictly follow the path, whereas goal-driven high-level instructions [43,57] mainly describe the target place and object and command the agent to retrieve a particular remote object. The embodied question answering task asks an agent to navigate and answer a question [12,55], and vision-and-language dialog [13,38,39,50] tasks use interactive communications with agents to guide the navigation. Due to the inherent multimodal nature, works on VLN tasks provide appealing and creative model architectures, such as cross-modal attention mechanism [16,34,35,49,52], awareness of objects [20,37,42], sequence modeling using transformers [8,22,40], Bayesian state tracking [1], and graph-based structures for better exploration [9,14,51]. The models are usually pretrained in a teacher forcing manner [8,19] and then fine-tuned using student forcing [3,9], stochastic sampling [31], or reinforcement learning [8,22,49,53]. While most existing VLN works focus on discrete environment with predefined navigation graphs, VLN in continuous environments [26,27] is

more practical in real world [2]. The discrete environments also prove to be beneficial for continuous VLN such as providing waypoint supervision [46] and enabling hierarchical modeling of low-level and high-level actions [7,21].

Data-centric VLN approaches. One of the major challenges of VLN remains the scarcity of training data, leading to a large gap between seen and unseen environments. Data augmentation is one effective approach to address over-fitting, such as to dropout environment features [49], change image styles [30], mixup rooms in different environments [33], generate more path-instruction pairs from speaker models [16,17,49] and new images from GANs [24]. However, those data augmentation methods are still based on a limited number of environments, abating generalization ability on unseen environments. Contrary to the above, VLN Bert [36] and Airbert [18] exploit abundant web image-captions to improve generalization. Nevertheless, such data do not fully resemble real navigation experience and thus they can only be used to train a compatibility model to measure path-instruction similarity instead of learning a navigation policy. In our work, photo-realistic 3D environments are used to learn a navigation policy.

3D environments. 3D environments and simulation platforms [25,29,41,47] are a basic foundation to promote research in embodied intelligence. There are artificial environments based on video game engines such as iThor [25] and VirtualHome [41], which utilize synthetic scenes and allow interactions with objects. However, synthetic scenes have limited visual diversity and do not fully reflect the real world. Thus, we focus on photo-realistic 3D environments for VLN. The MP3D [4] dataset is a collection of 90 labeled environments and is most widely adopted in existing VLN datasets such as R2R [3], RxR [28], REVERIE [43] and SOON [57]. Though possessing high-quality reconstructions, the number of houses in MP3D is limited. The Gibson [54] dataset contains 571 scenes, however they have poor 3D reconstruction quality. The recent HM3D dataset [45] has the largest number of 3D environments though with no labels. It contains 1,000 high-quality and diverse building-scale reconstructions (900 publicly released) around the world such as multi-floor residences, stores, offices, *etc.* In this work, we employ the unlabeled HM3D dataset to scale up VLN datasets.

3 Generating VLN Dataset from Unlabeled 3D Buildings

In this section, we present our approach to automatically generate large-scale VLN data from unlabeled 3D environments, specifically HM3D [45]. We consider a discrete navigation environment [3,43,57], which is the mainstream setting for VLN. The discrete setup treats each environment as an undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where \mathcal{V} denotes navigable nodes, and \mathcal{E} denotes connectivity edges. Each node $V_i \in \mathcal{V}$ corresponds to a panoramic RGB-D image captured at its location (x, y, z) . An agent is equipped with a camera and GPS sensors and moves between connected nodes on this navigation graph.

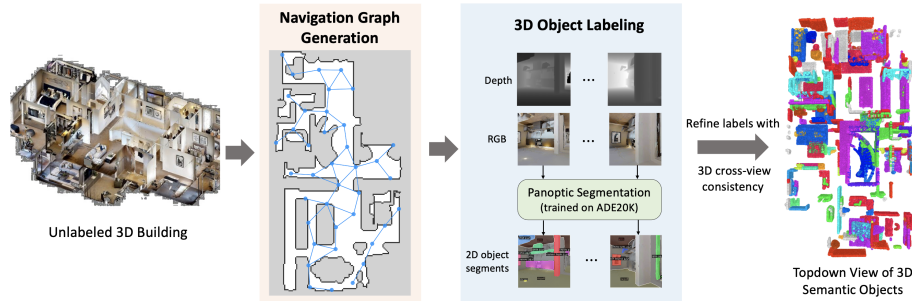


Fig. 2: We automatically construct the navigation graph and label 3D objects for each unlabeled 3D building.

We aim to generate VLN data with high-level instructions (similar to [43,57]), which mainly describe the target scene and specific object instance. The success of the agent is defined as arriving at a node where the target object is close and visible and correctly localizing the object. Therefore, the training data should consist of a language instruction, an initial position and orientation of the agent, a ground-truth trajectory to the target location and the position of the target object. In order to generate such training data, it is essential to know object locations in the 3D space instead of bounding boxes in 2D images. Otherwise, it can be hard to infer relationships between objects in different views leading to unreliable target locations. Moreover, a sentence that refers to a specific object should be generated since the object name alone is not specific. Generation of such data is challenging because the original 3D buildings come without any labels. Therefore, we propose to use image and text models pretrained on large-scale external datasets for pseudo labeling. This involves two stages: (i) building-level pre-processing to obtain navigation graphs and 3D object annotations (Sec. 3.1); and (ii) object-trajectory-instruction triplet generation (Sec. 3.2).

3.1 Building-level Pre-processing

We utilize the Habitat simulator [47] to load 3D buildings in HM3D [45] such that agents can navigate. This is a continuous navigation environment, thus we first convert each environment into a discrete navigation graph and then extract 3D pseudo object annotations for each building as illustrated in Fig. 2.

Constructing navigation graphs. We follow the characteristics of navigation graphs in the MP3D simulator [3] to create new graphs for continuous HM3D. For example, the average distance between connected nodes is around 2 meters and connectivity is defined by whether two nodes are visible and navigable from each other. However, the MP3D simulator relies on human efforts to create such graphs where the nodes are selected and navigability is inspected manually. Instead, we build the graph in a fully automatic way. We first randomly sample

20,000 navigable locations in the 3D environment and then greedily add locations as new nodes into the graph. The newly added location is the nearest one to existing nodes in the graph among all remaining candidates that are more than 2 meters away from the existing nodes. After collecting all the nodes, we use two criteria to connect different nodes: (i) the geodesic distance between nodes is below 3 meters; and (ii) the average distance of the depth image captured from one point to another should be larger than 2 meters. Fig. 2 (left) presents an example of navigation graph generation on one floor of the environment. For each node, we extract 36 RGB-D images from different orientations to represent the panoramic view, following the VLN setting [3].

Labeling 3D objects from 2D predictions. Since existing 3D datasets such as Scannet [11] are small and contain a limited number of object classes, pre-trained 3D object detectors are incapable of providing high-quality 3D object labels for HM3D which covers a diverse set of scenes and objects. Therefore, we propose to use an existing 2D image model to label 3D objects. Specifically, we use a panoptic segmentation model Mask2Former [10] trained on ADE20K [56] to generate instance masks for all images in the constructed graph. We project 2D pixel-wise semantic predictions into 3D point clouds using the camera intrinsic parameters and depth information, and thus obtain 3D bounding boxes. However, due to the domain gap between existing 2D labeled images and images from HM3D, the predictions of 2D models can be noisy as shown in Fig. 5. Even worse, a 3D object is often partially observed from one view, making the estimated 3D bounding box from a single location less accurate. In order to further improve 3D object labeling, we take advantage of cross-view consistency of 3D objects and merge 2D predictions from multiple views. To ease computation and reduce label noise, we downsample the extracted point clouds with class probabilities into voxels of size $0.1 \times 0.1 \times 0.1m^3$ in a similar way as semantic map construction [5]. Hence, 2D predictions from different views of an object can be merged together. We average class probabilities of all points inside each voxel and take the label with maximum probability to refine semantic prediction. The neighboring voxels of the same class are grouped together as a complete 3D object. In this way, we generate 3D objects for the whole building and also map 2D objects of different views into the unified 3D objects. This enables us to create reliable goal locations for target objects in the VLN task. Fig. 2 (right) shows example predictions of 3D semantic objects from a top-down view.

3.2 Data Generation: VLN Training Triplets

Based on the pseudo 3D object labels of the building, we generate object-trajectory-instruction triplets as shown in Fig. 3. For each 3D object, we obtain the corresponding goal locations in the graph where the object is visible and located within d_o meters. Then we randomly select a start node that is 4 to 9 steps away from the target location, and use the navigation graph to compute the shortest path as the expert trajectory. The final panoramic image in the trajectory is used to generate high-level VLN instructions.

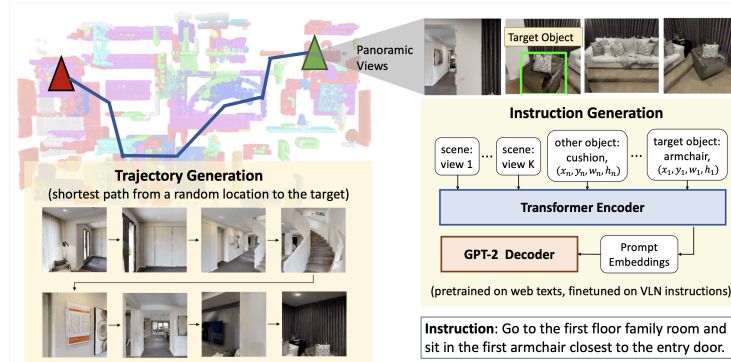


Fig. 3: To generate VLN training triplets, we first select a target object using the pseudo 3D object labels and then sample a trajectory using the navigation graph. Finally, we use a trained speaker model to generate high-level VLN instructions.

Generating instructions with object prompts. Previous works [16, 49] train speaker models from scratch on VLN datasets to generate instructions from an image trajectory. Due to the small amount of training examples in VLN datasets, such speaker models may suffer from inaccurate classification and are also unable to generate novel words beyond the training vocabulary. As a result, they even perform worse than template-based instructions on unseen images with diverse objects [18]. We alleviate these problems by (i) using object labels obtained from the 3D object labeling as prompts for the speaker model; and (ii) fine-tuning a pretrained large language model (GPT-2 [44]).

To be specific, our speaker model consists of a transformer encoder that produces prompt embeddings and a language decoder that generates sentences conditioning on the prompts. The encoder is fed with three types of tokens: target object token, other object tokens in the panorama, and view image tokens. The object token embedding encodes the object label, location, size, and visual representation, while the view image embedding is obtained from the visual representation and orientation embedding. A multi-layer transformer is adopted to learn relations of different tokens in order to generate more accurate referring expression for the target object. We average the output embeddings for each type of tokens separately, resulting in three tokens (target object, other objects, and view image) as prompts to the decoder. The decoder is initialized with pretrained GPT-2 [44], a state-of-the-art language model. It takes the above prompts as the first three tokens and sequentially predicts words. We finetune the speaker model end-to-end on the REVERIE dataset [43].

3.3 HM3D-AutoVLN: A Large-scale VLN Dataset

We employ the above procedure to process all unlabeled buildings in HM3D and generate a large-scale VLN dataset with high-level instructions, named HM3D-

Table 1: Statistics of training data on different VLN datasets. (gt.obj denotes whether the data relies on groundtruth object annotations)

dataset	gt.obj	#env.	#objs	#instr	#vocab	inst. length
REVERIE [43]	✓	60	12,029	10,466	1,140	18.64
REVERIE-Speaker [9]	✓	64	12,062	19,636	399	15.20
SOON [57]	✓	34	4,358	2,780	735	44.09
HM3D-AutoVLN	×	900	172,000	217,703	1,696	20.52

AutoVLN. Table 1 compares HM3D-AutoVLN with existing VLN datasets that also use high-level instructions. REVERIE and SOON are manually annotated datasets and REVERIE-Speaker is an augmented dataset generated by a speaker model trained with groundtruth room and object annotations. Our HM3D-AutoVLN is an order of magnitude larger than these datasets, in terms of environments, objects, and instructions. Due to the pretrained language model, our dataset also features a diverse vocabulary³ in the generated instructions.

4 VLN Model and Training

In this section, we briefly introduce a state-of-the-art VLN model DUET [9] and then describe its training procedure.

4.1 DUET VLN Model

The main idea of the DUal scaleE graph Transformer (DUET) [9] model is to build a topological map during navigation, which allows the model to make efficient long-term navigation plans over all navigable nodes in the graph instead of being limited to the neighboring nodes. The model consists of two modules: topological mapping and global action planning. The topological mapping module gradually adds newly observed locations to the map and updates node representations. The global action planning module is a dual-scale graph transformer to predict a next location in the map or a stop action at each step. The dual-scale architecture enables the model to jointly use fine-grained language grounding with fine-scale representations of the current location and global graph reasoning with coarse-scale representation of the map. The DUET model is the winning entry in ICCV 2021 REVERIE and SOON navigation challenge. We use the publicly released code⁴ in our experiments.

4.2 Training

Stage I: Pretraining on HM3D-AutoVLN. We use three proxy tasks to pretrain DUET [9], including Masked Language Modeling (MLM), Single-step

³ In Table 1 vocab, we only count words that occur more than 5 times in the dataset.

⁴ <https://github.com/cshizhe/VLN-DUET>

Action Prediction (SAP) and Object Grounding (OG). For MLM, the inputs are a masked instruction and a full expert trajectory (V_1, \dots, V_T) . The goal is to recover the masked words from the cross-modal visual representation. For SAP, we ask the model to predict the next action given the instruction and its navigation history. We randomly select the navigation history as: (i) (V_1, \dots, V_T) with target action *STOP*; (ii) (V_1, \dots, V_t) where $1 \leq t < T$ with target action V_{t+1} ; and (iii) a random node V_i in the graph, and choose a target node V_{i+1} with shortest overall distance from V_i to V_{i+1} and from V_{i+1} to V_T among all nodes in the constructed map. Finally, for OG, the model is fed with an instruction and expert trajectory (V_1, \dots, V_T) to predict the ground-truth object in V_T .

Stage II: Fine-tuning on downstream VLN datasets. We fine-tune a pre-trained DUE on each downstream task by using the pseudo interactive demonstrator algorithm [9] – a special case of student forcing that addresses exposure bias. We suggest two strategies for fine-tuning: (i) fine-tuning on the downstream dataset only; or (ii) balancing the size of HM3D-AutoVLN and the downstream dataset, and combining the two datasets for joint training. The latter strategy is successful at preventing catastrophic forgetting of the previously learned policy as shown in our experiments in Table 2.

5 Experiments

5.1 Experiment Setup

Datasets. We evaluate models on two VLN tasks with high-level instructions, REVERIE [43] and SOON [57]. Each dataset is divided into training, val seen, val unseen and a hidden test unseen split. The statistics of training splits are presented in Table 1. The REVERIE dataset provides groundtruth object bounding boxes at each location and only requires an agent to select one of the bounding boxes. The shortest path from the agent’s initial location to the target location is between 4 to 7 steps. The SOON dataset instead does not give groundtruth bounding boxes, and asks an agent to directly predict the orientation of the object’s center. The path lengths are also longer than REVERIE with 9.5 steps on average. Due to the increased task difficulty and smaller training dataset size, it is more challenging to achieve high performance on SOON than on REVERIE.

Evaluation metrics. We measure two types of metrics for the VLN task, navigation-only and remote object grounding. The navigation-only metrics focus solely on whether an agent arrives at the target location. We use standard navigation metrics [3] including **Success Rate (SR)**, the percentage of paths with the *final* location near any target locations within 3 meters; **Oracle Success Rate (OSR)**, the percentage of paths with *any* location close to target within 3 meters; and **SR weighted by Path Length (SPL)** which multiplies SR with the ratio between the length of shortest path and the agent’s predicted

path. As SPL takes into account navigation accuracy and efficiency, it is the primary metric in navigation. The remote object grounding metrics [43] consider both navigation and object grounding performance. The standard metrics are **Remote Grounding Success (RGS)** and **RGS weighted by Path Length (RGSPL)**. RGS in REVERIE is defined as correctly selecting the object among groundtruth bounding boxes, while in SOON as predicting a center point that is inside of the groundtruth polygon. RGSPL penalizes RGS by path length similar to SPL. For all these metrics, higher is better.

Implementation details. We adopt ViT-B/16 [15] pretrained on ImageNet to extract view and object features. For SOON dataset, we use the Mask2Former trained on ADE20K [10] to extract object bounding boxes and transfer the prediction in the same way as in REVERIE. We use the same hyper-parameters in modeling as the DUET model [9]. All experiments were run on a single Nvidia RTX8000 GPU. The best epoch is selected based on SPL on the val unseen split.

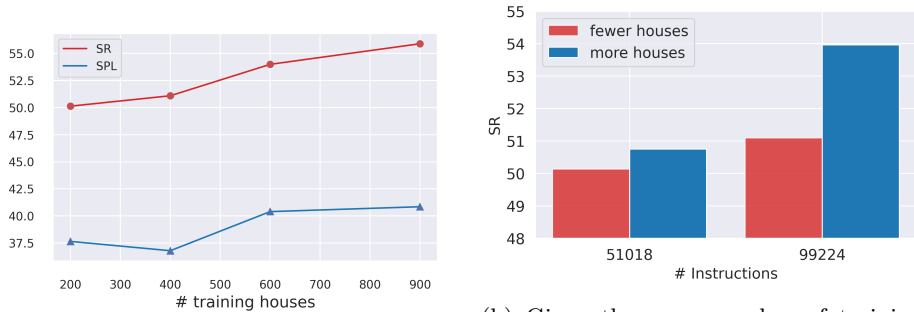
5.2 Ablation Studies

The main objective of our work is to explore how much VLN agents can benefit from large-scale synthesized dataset. In this section, we carry out extensive ablations on the REVERIE dataset to study the effectiveness of our HM3D-AutoVLN dataset and key design choices.

Contributions from HM3D-AutoVLN dataset. In Table 2, we compare the impact of VLN data generated from different sources for training DUET. Row 1 (R1) only relies on manually annotated instructions on 60 buildings in MP3D. Row 2 (R2) utilizes groundtruth room and object annotations to generate instructions for all objects in the 60 seen buildings. Such data is of high-quality and significantly improves VLN performance, *e.g.* 3% on SPL. In R3, we only use our generated HM3D-AutoVLN for pretraining, while still finetuning on the REVERIE train split. Due to the visual diversity of the 900 additional buildings, it brings a larger boost than R2. Compared to R1, the improvement of the SPL metric is 8.6%, whereas other metrics like SR improve by 6.5%. This suggests that the pretraining on large-scale environments enables the model to learn efficient exploration, even without any additional manual annotations. Moreover, when the model is fine-tuned jointly on the HM3D-AutoVLN and REVERIE datasets (R4), the SR metric is further improved by 5% over R3. This indicates that fine-tuning on the downstream dataset alone may suffer from forgetting and lead to worse generalization performance. Compared to the navigation metrics, the remote object grounding metrics see modest improvements. We hypothesize that though our generated instructions often describe the object and scene accurately, they are not discriminative enough to refer to a specific object in the environment (*e.g.* unable to discriminate between multiple pillows placed on a sofa) or confuse predicting relationships between objects (*e.g.* second pillow from

Table 2: DUET performance on REVERIE (RVR) val unseen split using different training data. [†] denotes manual object annotations are used to synthesize data

	Training Data		Navigation			Grounding	
	Pretrain	Finetune	OSR	SR	SPL	RGS	RG SPL
R1	RVR	RVR	48.74	44.36	30.79	30.30	21.08
R2	RVR+Speaker [†]	RVR	51.07	46.98	33.73	32.15	23.03
R3	HM3D	RVR	54.81	50.87	39.36	34.65	26.79
R4	HM3D	RVR+HM3D	62.14	55.89	40.85	36.58	26.76



(a) Navigation performance with respect to the number of training environments in HM3D-AutoVLN dataset.

(b) Given the same number of training examples, collecting data from more environments (blue) performs better than fewer environments.

Fig. 4: Influence of the number of environments on DUET performance.

the left). We see improving RGS and RGSPL as a promising future direction that would need to take into account relations between objects.

Impact of the number of training environments. Here, we evaluate the impact of the number of training environments. As pretraining on HM3D-AutoVLN mostly improves navigation performance, we mainly show navigation metrics SR and SPL. The full result table is provided in the supplementary material. As shown in Figure 4a, more environments continuously improve the navigation performance. We observe that even with the full 900 environments in HM3D, the gains are not saturated but increase gradually. We also evaluate the impact of the number of environments given a fixed budget of VLN training examples. Figure 4b shows that using more environments improves the performance. On the **left**, the red bar uses 200 environments with 51,018 instructions, and the blue bar use the same amount of instructions but covers all 900 environments in HM3D. The **right** part is the same, but uses 400 environments for the red bar. The results suggest that it is beneficial to increase the number of environments rather than just increasing the trajectories for a limited number of environments.

Table 3: DUET performance when using a fraction of the supervised data

HM3D Pretrain	#environments	#instructions	Navigation			Grounding	
			OSR	SR	SPL	RGS	RG SPL
×	60	10,466	48.74	44.36	30.79	30.30	21.08
✓	0	0	43.08	36.81	25.28	20.82	13.74
✓	1	449	50.78	42.12	29.55	25.02	17.26
✓	10	1,404	50.47	43.79	33.61	26.30	20.24
✓	30	5,244	60.81	53.71	39.26	34.42	25.11
✓	60	10,466	62.14	55.89	40.85	36.58	26.76

Finetuning with a small amount of supervised data. A large-scale automatically generated dataset improves pretraining and hence can reduce the supervised data requirement in downstream tasks. We verify the effectiveness of HM3D-AutoVLN on a few-shot learning setting [18] that compares the impact of finetuning on a variable number of REVERIE environments, see Table 3. Even without finetuning on any supervised instructions from REVERIE, our pre-trained model achieves fair performance in comparison to a baseline that is only finetuned on all supervised data. On the SPL metric, we outperform the baseline DUET model when finetuning on 10 environments (1/6 of the supervised data). Moreover, finetuning with half the original data (30 environments) achieves significant boosts on all metrics compared to the baseline model. A similar trend can be observed on the SOON dataset, see supplementary material.

Comparing distance to objects. As mentioned in Sec. 3.2, we select some of the visible objects that are close to the agent, within d_o meters, to generate instructions. In Table 4, we compare the influence of different distances to the objects on the VLN performance. Larger d_o allows us to generate more instructions. We can see that while including additional remote objects increases the number of instructions, it leads to a small drop in performance as the model struggles to identify objects that are small and far away.

Table 4: DUET performance on instructions where the visible objects are at a different distance d_o from the agent location

d_o	#instrs	Navigation			Grounding	
		OSR	SR	SPL	RGS	RG SPL
2	217,703	62.14	55.89	40.85	36.58	26.76
3	396,401	57.37	53.25	40.38	34.28	25.65
∞	544,606	59.98	53.37	38.03	35.70	25.50

Table 5: Comparison of different speaker models in terms of manual captioning evaluation and the followup navigation performance

	Captioning			Navigation	
	Room	Obj	Rel	SR	RGS
Template	0.13	0.76	0.05	52.20	32.75
LSTM	0.58	0.65	0.27	49.59	32.29
GPT2 (Ours)	0.73	0.78	0.35	55.89	36.58

Evaluating quality of the HM3D-AutoVLN dataset. We validate each annotation procedure in our automatic dataset construction. For *navigation graph*



Fig. 5: Qualitative examples of pseudo labelling

generation, we measure whether the graph covers the whole building. Assuming each navigation node covers a circle with a radius of 2 meters, the graph achieves a high coverage rate of 93.4% on average. For *3D object labeling*, we randomly select 300 bounding boxes and manually annotate semantic labels for them. We observe that 37.4% of them are correctly predicted with 2D predictions, whereas 58.3% of them were correctly predicted when applying cross-view consistency, showing an absolute improvement of 21%. Examples in Figure 5 highlight the advantages of using cross-view consistency. Due to the distorted view, it’s reasonable that the 2D models wrongly predict the mirror and wardrobe to be a window and door respectively. Cross-view consistency also helps to integrate scene context, swapping a table to a desk and a normal chair to a swivel chair. For *instruction generation*, we further compare our GPT2-based speaker model with template-based methods and a LSTM baseline [9]. We manually evaluate 100 randomly selected instructions by measuring whether the instruction correctly mentions the target room, object class and object instance (with correct relations). We also measure the VLN performance using the generated instructions. Our model performs best on manual evaluation and on downstream VLN tasks as shown in Table 5.

5.3 Comparison with State of the Art

In Table 6, we compare with the state of the art on the REVERIE dataset. To demonstrate the contributions of our automatically constructed dataset, we further train additional VLN agents with the augmentation of the HM3D-AutoVLN dataset, including EnvDrop [49], RecBert [22] and HAMT [8]. Our proposed dataset improves results of all methods and gives a particularly large boost to high-capacity models. When pretraining DUET on HM3D-AutoVLN, the increase in performance over DUET with pretraining is significant for all metrics on both the val unseen and the test unseen splits. For example, the SPL measure increases by 7.1% and 2.8% on val unseen and test unseen splits of REVERIE. Table 7 provides results for the SOON dataset. Note that instructions from the SOON dataset are not used to train our speaker model (*c.f.* Sec. 3.2) and are somewhat different from the REVERIE instructions. Nevertheless, the large performance increase demonstrates cross-domain benefits of pretraining on an

Table 6: Comparison with the state-of-the-art methods on REVERIE dataset

Methods	Val Unseen					Test Unseen				
	Navigation		Grounding			Navigation		Grounding		
	OSR	SR	SPL	RGS	RGSP	OSR	SR	SPL	RGS	RGSP
Human	-	-	-	-	-	86.83	81.51	53.66	77.84	51.44
Seq2Seq [3]	8.07	4.20	2.84	2.16	1.63	6.88	3.99	3.09	2.00	1.58
RCM [53]	14.23	9.29	6.97	4.89	3.89	11.68	7.84	6.67	3.67	3.14
SMNA [34]	11.28	8.15	6.44	4.54	3.61	8.39	5.80	4.53	3.10	2.39
FAST-MAttNet [43]	28.20	14.40	7.19	7.84	4.67	30.63	19.88	11.61	11.28	6.08
SIA [32]	44.67	31.53	16.28	22.41	11.56	44.56	30.80	14.85	19.02	9.20
Airbert [18]	34.51	27.89	21.88	18.23	14.18	34.20	30.28	23.61	16.83	13.28
EnvDrop [49]	26.3	23.0	19.9	-	-	-	-	-	-	-
+HM3D-AutoVLN	29.3	25.2	20.6	-	-	-	-	-	-	-
RecBERT (oscar) [22]	27.66	25.53	21.06	14.20	12.00	26.67	24.62	19.48	12.65	10.00
+HM3D-AutoVLN	33.23	29.20	23.12	18.12	14.18	-	-	-	-	-
HAMT [8]	36.84	32.95	30.20	18.92	17.28	33.41	30.40	26.67	14.88	13.08
+HM3D-AutoVLN	42.09	37.80	31.35	23.03	18.88	-	-	-	-	-
DUET [9]	51.07	46.98	33.73	32.15	23.03	56.91	52.51	36.06	31.88	22.06
+HM3D-AutoVLN	62.14	55.89	40.85	36.58	26.76	62.3	55.17	38.88	32.23	22.68

Table 7: Comparison with the state-of-the-art methods on the SOON dataset

Methods	Val Unseen				Test Unseen			
	OSR	SR	SPL	RGSP	OSR	SR	SPL	RGSP
GBE [57]	28.54	19.52	13.34	1.16	21.45	12.90	9.23	0.45
DUET [9]	50.91	36.28	22.58	3.75	43.00	33.44	21.42	4.17
DUET (+HM3D)	53.19	41.00	30.69	4.06	48.74	40.36	27.83	5.11

automatically collected large-scale dataset. Finally, we can also observe that the gain for object grounding is less significant, as discussed before this can be explained by the confusion between objects of the same categories and erroneous spatial relations.

6 Conclusion

This work addresses the lack of training data for VLN tasks. We propose to automatically generate pseudo 3D object labels and VLN instructions for a collection of large-scale unlabeled 3D environments. Training on our new dataset HM3D-AutoVLN significantly improves VLN performance due to the large-scale pretraining or co-training. It also provides insights on the importance of dataset collection and the challenges inherent to leveraging unlabeled environments. In particular, it shows that the diversity of navigation environments is more important than the number of training samples alone.

Acknowledgements. This work was granted access to the HPC resources of IDRIS under the allocation 101002 made by GENCI. This work is funded in part by the French government under management of Agence Nationale de la Recherche as part of the “Investissements d’avenir” program, reference ANR19-P3IA-0001 (PRAIRIE 3IA Institute) and by Louis Vuitton ENS Chair on Artificial Intelligence.

References

1. Anderson, P., Shrivastava, A., Parikh, D., Batra, D., Lee, S.: Chasing ghosts: Instruction following as bayesian state tracking. *NeurIPS* **32** (2019) [3](#)
2. Anderson, P., Shrivastava, A., Truong, J., Majumdar, A., Parikh, D., Batra, D., Lee, S.: Sim-to-real transfer for vision-and-language navigation. In: *CoRL*. pp. 671–681. PMLR (2021) [4](#)
3. Anderson, P., Wu, Q., Teney, D., Bruce, J., Johnson, M., Sünderhauf, N., Reid, I., Gould, S., Van Den Hengel, A.: Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In: *CVPR*. pp. 3674–3683 (2018) [1](#), [3](#), [4](#), [5](#), [6](#), [9](#), [14](#)
4. Chang, A., Dai, A., Funkhouser, T., Halber, M., Niebner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3d: Learning from rgb-d data in indoor environments. In: *3DV*. pp. 667–676. IEEE (2017) [2](#), [4](#)
5. Chaplot, D.S., Gandhi, D.P., Gupta, A., Salakhutdinov, R.R.: Object goal navigation using goal-oriented semantic exploration. *NeurIPS* **33**, 4247–4258 (2020) [6](#)
6. Chen, H., Suhr, A., Misra, D., Snavely, N., Artzi, Y.: Touchdown: Natural language navigation and spatial reasoning in visual street environments. In: *CVPR*. pp. 12538–12547 (2019) [3](#)
7. Chen, K., Chen, J.K., Chuang, J., Vázquez, M., Savarese, S.: Topological planning with transformers for vision-and-language navigation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 11276–11286 (2021) [4](#)
8. Chen, S., Guhur, P.L., Schmid, C., Laptev, I.: History aware multimodal transformer for vision-and-language navigation. In: *NeurIPS* (2021) [3](#), [13](#), [14](#)
9. Chen, S., Guhur, P.L., Tapaswi, M., Schmid, C., Laptev, I.: Think global, act local: Dual-scale graph transformer for vision-and-language navigation. In: *CVPR* (2022) [3](#), [8](#), [9](#), [10](#), [13](#), [14](#)
10. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation. *arXiv* (2021) [3](#), [6](#), [10](#)
11. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: *CVPR*. pp. 5828–5839 (2017) [6](#)
12. Das, A., Datta, S., Gkioxari, G., Lee, S., Parikh, D., Batra, D.: Embodied question answering. In: *CVPR*. pp. 1–10 (2018) [3](#)
13. De Vries, H., Shuster, K., Batra, D., Parikh, D., Weston, J., Kiela, D.: Talk the walk: Navigating new york city through grounded dialogue. *arXiv preprint arXiv:1807.03367* (2018) [3](#)
14. Deng, Z., Narasimhan, K., Russakovsky, O.: Evolving graphical planner: Contextual global planning for vision-and-language navigation. In: *NeurIPS*. vol. 33 (2020) [3](#)
15. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16× 16 words: Transformers for image recognition at scale. *ICLR* (2020) [10](#)
16. Fried, D., Hu, R., Cirik, V., Rohrbach, A., Andreas, J., Morency, L.P., Berg-Kirkpatrick, T., Saenko, K., Klein, D., Darrell, T.: Speaker-follower models for vision-and-language navigation. In: *NeurIPS*. pp. 3318–3329 (2018) [2](#), [3](#), [4](#), [7](#)
17. Fu, T.J., Wang, X.E., Peterson, M.F., Grafton, S.T., Eckstein, M.P., Wang, W.Y.: Counterfactual vision-and-language navigation via adversarial path sampler. In: *ECCV*. pp. 71–86. Springer (2020) [4](#)

18. Guhur, P.L., Tapaswi, M., Chen, S., Laptev, I., Schmid, C.: Airbert: In-domain pretraining for vision-and-language navigation. In: ICCV. pp. 1634–1643 (2021) [2](#), [4](#), [7](#), [12](#), [14](#), [19](#)
19. Hao, W., Li, C., Li, X., Carin, L., Gao, J.: Towards learning a generic agent for vision-and-language navigation via pre-training. In: CVPR. pp. 13137–13146 (2020) [3](#)
20. Hong, Y., Rodriguez, C., Qi, Y., Wu, Q., Gould, S.: Language and visual entity relationship graph for agent navigation. *NeurIPS* **33**, 7685–7696 (2020) [3](#)
21. Hong, Y., Wang, Z., Wu, Q., Gould, S.: Bridging the gap between learning in discrete and continuous environments for vision-and-language navigation. In: CVPR. pp. 15439–15449 (2022) [4](#)
22. Hong, Y., Wu, Q., Qi, Y., Rodriguez-Opazo, C., Gould, S.: Vln BERT: A recurrent vision-and-language BERT for navigation. In: CVPR. pp. 1643–1653 (2021) [3](#), [13](#), [14](#)
23. Irshad, M.Z., Ma, C., Kira, Z.: Hierarchical cross-modal agent for robotics vision-and-language navigation. In: ICRA. pp. 13238–13246 (2021) [3](#)
24. Koh, J.Y., Lee, H., Yang, Y., Baldrige, J., Anderson, P.: Pathdreamer: A world model for indoor navigation. In: ICCV. pp. 14738–14748 (2021) [4](#)
25. Kolve, E., Mottaghi, R., Han, W., VanderBilt, E., Weihs, L., Herrasti, A., Gordon, D., Zhu, Y., Gupta, A.K., Farhadi, A.: AI2-THOR: An interactive 3d environment for visual ai. *ArXiv abs/1712.05474* (2017) [4](#)
26. Krantz, J., Gokaslan, A., Batra, D., Lee, S., Maksymets, O.: Waypoint models for instruction-guided navigation in continuous environments. In: ICCV. pp. 15162–15171 (2021) [3](#)
27. Krantz, J., Wijmans, E., Majumdar, A., Batra, D., Lee, S.: Beyond the nav-graph: Vision-and-language navigation in continuous environments. In: ECCV. pp. 104–120. Springer (2020) [3](#)
28. Ku, A., Anderson, P., Patel, R., Ie, E., Baldrige, J.: Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In: EMNLP. pp. 4392–4412 (2020) [1](#), [3](#), [4](#)
29. Li, C., Xia, F., Martín-Martín, R., Lingelbach, M., Srivastava, S., Shen, B., Vainio, K.E., Gokmen, C., Dharan, G., Jain, T., et al.: igibson 2.0: Object-centric simulation for robot learning of everyday household tasks. In: CoRL. pp. 455–465. PMLR (2022) [4](#)
30. Li, J., Tan, H., Bansal, M.: Envedit: Environment editing for vision-and-language navigation. In: CVPR. pp. 15407–15417 (2022) [2](#), [4](#)
31. Li, X., Li, C., Xia, Q., Bisk, Y., Celikyilmaz, A., Gao, J., Smith, N.A., Choi, Y.: Robust navigation with language pretraining and stochastic sampling. In: EMNLP. pp. 1494–1499 (2019) [3](#)
32. Lin, X., Li, G., Yu, Y.: Scene-intuitive agent for remote embodied visual grounding. In: CVPR. pp. 7036–7045 (2021) [14](#)
33. Liu, C., Zhu, F., Chang, X., Liang, X., Ge, Z., Shen, Y.D.: Vision-language navigation with random environmental mixup. In: ICCV. pp. 1644–1654 (2021) [2](#), [4](#)
34. Ma, C.Y., Lu, J., Wu, Z., AlRegib, G., Kira, Z., Socher, R., Xiong, C.: Self-monitoring navigation agent via auxiliary progress estimation. In: ICLR (2019) [3](#), [14](#)
35. Ma, C.Y., Wu, Z., AlRegib, G., Xiong, C., Kira, Z.: The regretful agent: Heuristic-aided navigation through progress estimation. In: CVPR. pp. 6732–6740 (2019) [3](#)

36. Majumdar, A., Shrivastava, A., Lee, S., Anderson, P., Parikh, D., Batra, D.: Improving vision-and-language navigation with image-text pairs from the web. In: ECCV. pp. 259–274. Springer (2020) [2](#), [4](#)
37. Moudgil, A., Majumdar, A., Agrawal, H., Lee, S., Batra, D.: Soat: A scene-and object-aware transformer for vision-and-language navigation. NeurIPS **34** (2021) [3](#)
38. Nguyen, K., Daumé III, H.: Help, ANNA! visual navigation with natural multi-modal assistance via retrospective curiosity-encouraging imitation learning. arXiv preprint arXiv:1909.01871 (2019) [3](#)
39. Padmakumar, A., Thomason, J., Shrivastava, A., Lange, P., Narayan-Chen, A., Gella, S., Piramuthu, R., Tur, G., Hakkani-Tur, D.: Teach: Task-driven embodied agents that chat. In: AAAI. vol. 36, pp. 2017–2025 (2022) [3](#)
40. Pashevich, A., Schmid, C., Sun, C.: Episodic transformer for vision-and-language navigation. In: ICCV. pp. 15942–15952 (2021) [3](#)
41. Puig, X., Ra, K., Boben, M., Li, J., Wang, T., Fidler, S., Torralba, A.: Virtual-home: Simulating household activities via programs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8494–8502 (2018) [4](#)
42. Qi, Y., Pan, Z., Hong, Y., Yang, M.H., van den Hengel, A., Wu, Q.: The road to know-where: An object-and-room informed sequential bert for indoor vision-language navigation. In: ICCV. pp. 1655–1664 (2021) [3](#)
43. Qi, Y., Wu, Q., Anderson, P., Wang, X., Wang, W.Y., Shen, C., Hengel, A.v.d.: Reverie: Remote embodied visual referring expression in real indoor environments. In: CVPR. pp. 9982–9991 (2020) [2](#), [3](#), [4](#), [5](#), [7](#), [8](#), [9](#), [10](#), [14](#)
44. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019) [3](#), [7](#)
45. Ramakrishnan, S.K., Gokaslan, A., Wijmans, E., Maksymets, O., Clegg, A., Turner, J., Undersander, E., Galuba, W., Westbury, A., Chang, A.X., et al.: Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. arXiv preprint arXiv:2109.08238 (2021) [1](#), [2](#), [3](#), [4](#), [5](#)
46. Raychaudhuri, S., Wani, S., Patel, S., Jain, U., Chang, A.: Language-aligned waypoint (law) supervision for vision-and-language navigation in continuous environments. In: EMNLP. pp. 4018–4028 (2021) [4](#)
47. Savva, M., Kadian, A., Maksymets, O., Zhao, Y., Wijmans, E., Jain, B., Straub, J., Liu, J., Koltun, V., Malik, J., et al.: Habitat: A platform for embodied ai research. In: ICCV. pp. 9339–9347 (2019) [4](#), [5](#)
48. Shridhar, M., Thomason, J., Gordon, D., Bisk, Y., Han, W., Mottaghi, R., Zettlemoyer, L., Fox, D.: Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In: CVPR. pp. 10740–10749 (2020) [3](#)
49. Tan, H., Yu, L., Bansal, M.: Learning to navigate unseen environments: Back translation with environmental dropout. In: NAACL. pp. 2610–2621 (2019) [2](#), [3](#), [4](#), [7](#), [13](#), [14](#)
50. Thomason, J., Murray, M., Cakmak, M., Zettlemoyer, L.: Vision-and-dialog navigation. In: CoRL. pp. 394–406. PMLR (2020) [3](#)
51. Wang, H., Wang, W., Liang, W., Xiong, C., Shen, J.: Structured scene memory for vision-language navigation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8455–8464 (2021) [3](#)
52. Wang, H., Wang, W., Shu, T., Liang, W., Shen, J.: Active visual information gathering for vision-language navigation. In: ECCV. pp. 307–322. Springer (2020) [3](#)

- 53. Wang, X., Huang, Q., Celikyilmaz, A., Gao, J., Shen, D., Wang, Y.F., Wang, W.Y., Zhang, L.: Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In: CVPR. pp. 6629–6638 (2019) [3](#), [14](#)
- 54. Xia, F., Zamir, A.R., He, Z., Sax, A., Malik, J., Savarese, S.: Gibson env: Real-world perception for embodied agents. In: CVPR. pp. 9068–9079 (2018) [4](#)
- 55. Yu, L., Chen, X., Gkioxari, G., Bansal, M., Berg, T.L., Batra, D.: Multi-target embodied question answering. In: CVPR. pp. 6309–6318 (2019) [3](#)
- 56. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: CVPR. pp. 633–641 (2017) [6](#)
- 57. Zhu, F., Liang, X., Zhu, Y., Yu, Q., Chang, X., Liang, X.: Soon: Scenario oriented object navigation with graph-based exploration. In: CVPR. pp. 12689–12699 (2021) [2](#), [3](#), [4](#), [5](#), [8](#), [9](#), [14](#)

We first present additional automatically generated examples from our HM3D-AutoVLN dataset in Sec. A. Then, we present further ablation analysis of our methods and also discuss results on the SOON dataset (equivalent analysis for the REVERIE dataset was in the main paper) in Sec. B. Finally, some qualitative visualizations of predicted navigation results are presented in Sec. C.

A HM3D-AutoVLN Dataset

In Fig. 6, we provide more examples of generated navigation graphs and object-trajectory-instruction triplets in the HM3D-AutoVLN dataset. Each trajectory consists of a sequence of panoramas. We only visualize the oriented view image at each step in Fig. 6 for simplicity.

B Additional Ablations

B.1 Comparison of Different Generated Instructions

We compare our generated instructions from the proposed speaker model (Sec. 3.2 of the main paper) with two types of template-based instructions. The first type of template instructions (Template.Obj) only describes the target object name, *e.g.*, “open the window”, “clean the mirror”. The second type (Template.Sent) uses sentence structures in REVERIE training split as templates [18] and replaces the noun phrases in the template with detected objects in the scene, *e.g.*, “go to bedroom and clean the chandelier”. As these template instructions are less accurate and contain less details, we generate three instructions per target object to alleviate the impact of noisy instructions. As shown in Table 8, even with fewer number of instructions per object, the generated instructions from our speaker model outperform the two template-based instructions.

Table 8: DUET performance on the REVERIE val unseen split using different methods for generating instructions.

Method	#Instrs	inst.len	Navigation			Grounding	
			OSR	SR	SPL	RGS	RGSPL
Template.Obj	653,109	3.88	54.33	49.28	37.66	29.85	23.07
Template.Sent	653,109	11.50	56.16	52.20	39.90	32.75	24.86
GPT2 (Ours)	217,703	20.52	62.14	55.89	40.85	36.58	26.76

B.2 Influence of the Number of Training Environments

Table 9 and Table 10 present the results for all evaluation metrics corresponding to the SR/SPL results in Fig. 4 of the main paper. We can see that more training environments are beneficial for the performance on unseen environments in

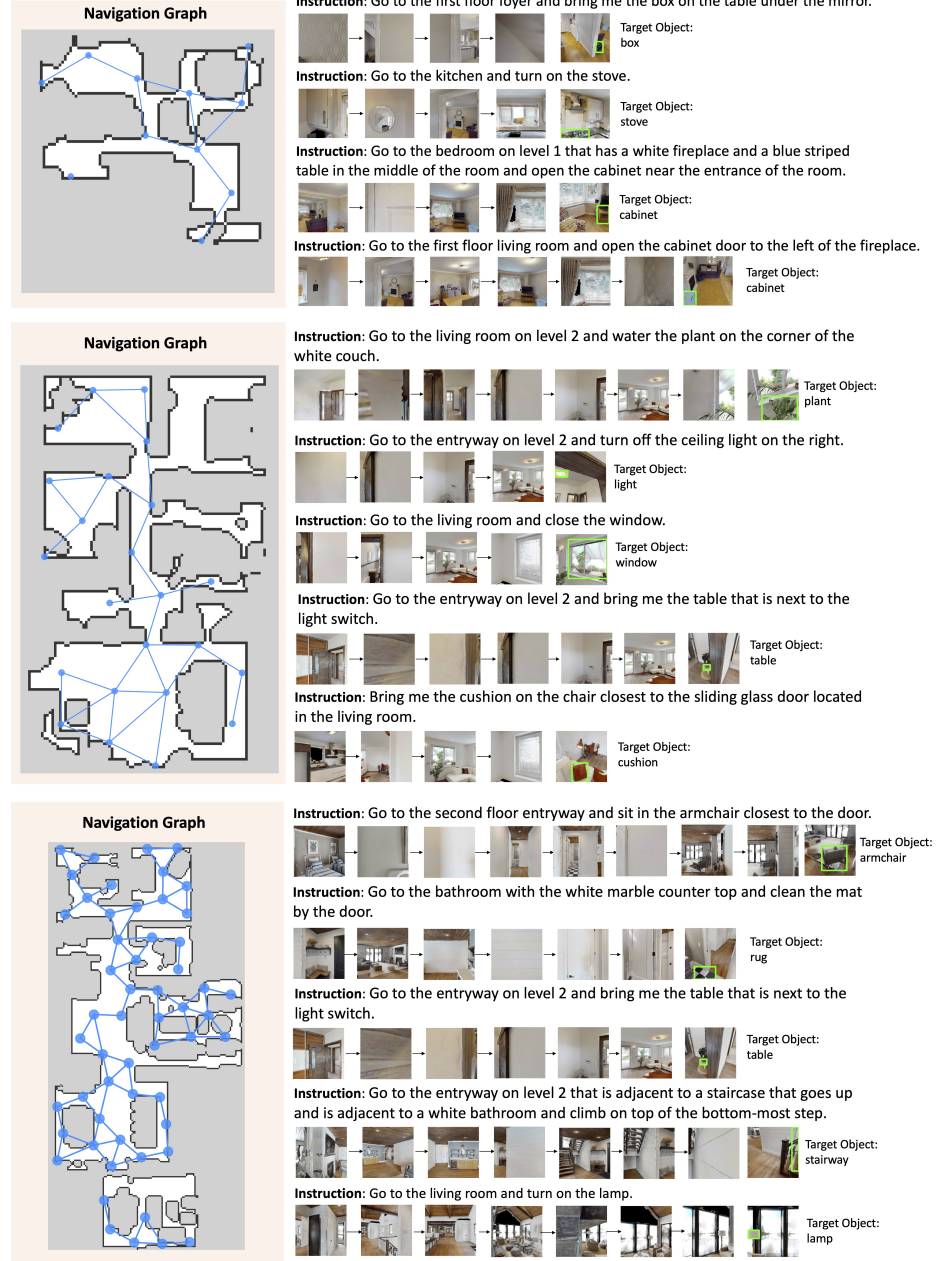


Fig. 6: Examples of generated navigation graphs and object-trajectory-instruction triplets in HM3D-AutoVLN dataset.

Table 9: DUET performance on the REVERIE val unseen split using different number of training environments from the HM3D-AutoVLN dataset.

#Envs	Navigation			Grounding	
	OSR	SR	SPL	RGS	RGSPL
200	55.61	50.13	37.65	32.92	24.68
400	55.04	51.09	36.79	32.43	23.16
600	60.61	53.99	40.40	33.85	25.52
900	62.14	55.89	40.85	36.58	26.76

Table 10: DUET performance on the REVERIE val unseen split using the same number of instructions but different number of training environments in HM3D-AutoVLN dataset (51,018 is the total number of instructions contained in the 200 training environments, and so is 99,224 in 400 training environments)

#Instrs	#Envs	Navigation			Grounding	
		OSR	SR	SPL	RGS	RGSPL
51,018	200	55.61	50.13	37.65	32.92	24.68
	900	54.05	50.75	39.52	32.35	25.22
99,224	400	55.04	51.09	36.79	32.43	23.16
	900	60.44	53.96	38.35	36.01	25.56

Table 9 and that this result is reflected across all metrics. When using a fixed number of instructions, it is better to collect training examples from more environments as shown in Table 10.

B.3 Few-shot Results

In Table 11, we show the performance of using different amounts of training data from the SOON dataset to fine-tune the DUET model pretrained on our HM3D-AutoVLN dataset. This is comparable to Table 3 of the main paper that presented results on the REVERIE dataset. To be noted, instructions in SOON dataset are not used to train our speaker model, and the style of instructions in HM3D-AutoVLN is quite different from the style in the SOON dataset. Despite the style differences, the pretraining allows our model to achieve better performance with a small amount of supervised data.

C Qualitative Results

Fig. 7 compares some predicted trajectories from DUET models with and without pretraining on our HM3D-AutoVLN dataset. We can see that the pretraining allows the navigation model to explore the unseen environment more efficiently.

Table 11: DUET performance on SOON val unseen split using different number of SOON training environments in fine-tuning

Pretrain	#Envs	#Instrs	Navigation			Grounding
			OSR	SR	SPL	RG SPL
×	34	2,780	50.91	36.28	22.58	3.75
✓	0	0	14.75	11.21	7.92	0.38
✓	1	67	37.32	25.22	17.97	2.17
✓	10	848	45.58	34.51	24.38	2.75
✓	34	2,780	53.19	41.00	30.69	4.06

Fig. 8 further presents some failure cases. The main problem is to correctly decide the stop location aligned to the fine-grained instruction. For example, in the last two examples in Fig. 8, the agent has navigated close to the target location but failed to stop and continued the exploration.

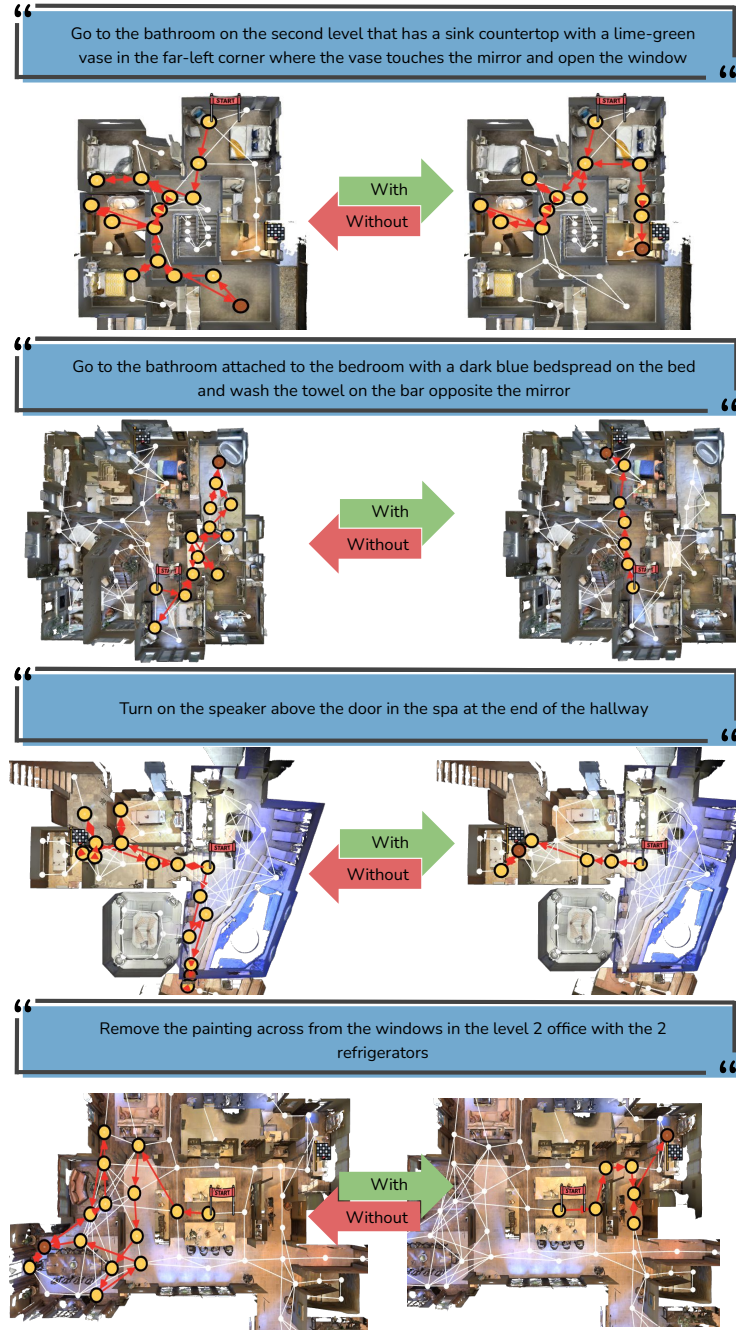


Fig. 7: Examples where pretraining with HM3D-AutoVLN improved the performance of DUET on REVERIE val unseen split. The checkered flag and maroon node denote the target and predicted destinations respectively.

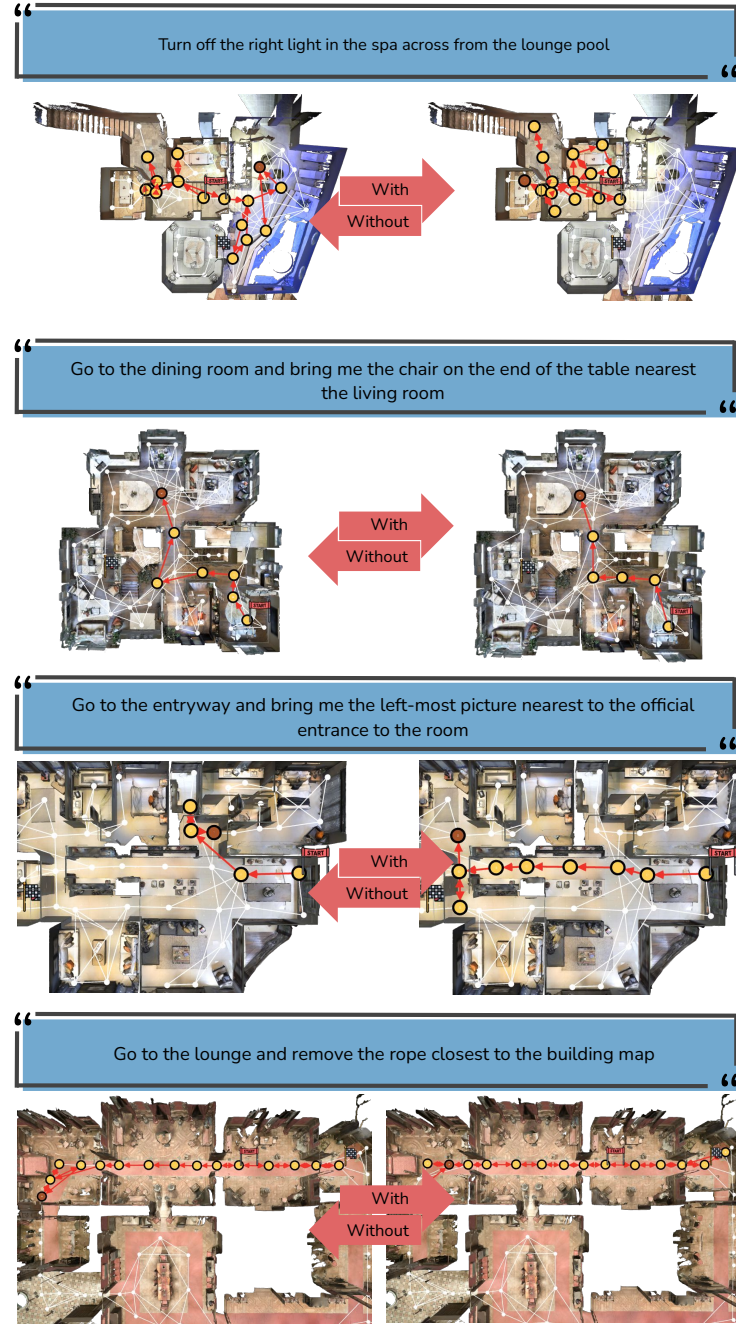


Fig. 8: Examples where DUET models with and without pretraining on HM3D-AutoVLN both fail on REVERIE val unseen split. The checkered flag and maroon node denote the target and predicted destinations respectively.