

Scaling Data Generation in Vision-and-Language Navigation

Zun Wang^{*1,2} Jialu Li^{*3} Yicong Hong^{*†1}
 Yi Wang² Qi Wu⁴ Mohit Bansal³ Stephen Gould¹ Hao Tan⁵ Yu Qiao²
¹The Australian National University ²OpenGVLab, Shanghai AI Laboratory
³UNC, Chapel Hill ⁴University of Adelaide ⁵Adobe Research

wangzun@pjlab.org.cn, jialuli@cs.unc.edu, mr.yiconghong@gmail.com

Project URL: <https://github.com/wz0919/ScaleVLN>

Abstract

Recent research in language-guided visual navigation has demonstrated a significant demand for the diversity of traversable environments and the quantity of supervision for training generalizable agents. To tackle the common data scarcity issue in existing vision-and-language navigation datasets, we propose an effective paradigm for generating large-scale data for learning, which applies 1200+ photo-realistic environments from HM3D and Gibson datasets and synthesizes 4.9 million instruction-trajectory pairs using fully-accessible resources on the web. Importantly, we investigate the influence of each component in this paradigm on the agent’s performance and study how to adequately apply the augmented data to pre-train and fine-tune an agent. Thanks to our large-scale dataset, the performance of an existing agent can be pushed up (+11% absolute with regard to previous SoTA) to a significantly new best of 80% single-run success rate on the R2R test split by simple imitation learning. The long-lasting generalization gap between navigating in seen and unseen environments is also reduced to less than 1% (versus 8% in the previous best method). Moreover, our paradigm also facilitates different models to achieve new state-of-the-art navigation results on CVDN, REVERIE, and R2R in continuous environments.

1. Introduction

Vision-and-Language Navigation (VLN) [10] is a challenging task that requires an agent to navigate in photo-realistic environments, following human natural language instructions such as “Walk downstairs, move towards the dining table, turn left to the kitchen, and stop in front

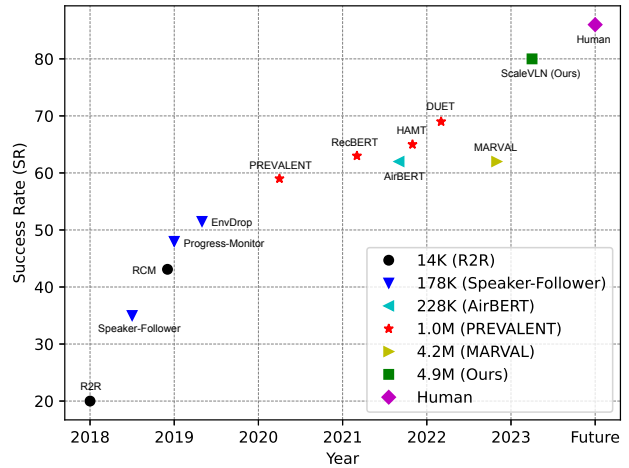


Figure 1: Agent success rate with increasing data size on addressing the R2R navigation task. Our proposed method creates 4.9M instruction-trajectories pairs for learning, which greatly boosts the agent’s performance, and for the first time approaching human results.

of the fridge.” Addressing VLN relies heavily on correctly interpreting the instructions, perceiving the environments, and learning from interaction, which demands a large amount of diverse visual-language data for learning. Recent research shows that scaling up the diversity of environments and the quantity of demonstration for training VLN agents are promising in improving generalization to unseen scenes [17, 37]. Compared to previous approaches of addressing data scarcity by augmenting agent’s observations [47, 74] or employing large vision-linguistic models pre-trained with image-text data from the web [27, 31, 54, 71, 72], utilizing additional traversable environments allows the agents to learn from in-domain visual-language data and physical interaction in the space.

In light of this, recent large datasets which contain hundreds of interactive scenes have been created [20, 66, 84], as well as a vast amount of human demonstrations have been

^{*}Equal contribution. [†]Project lead.

[♣]Research done during internship at Shanghai AI Lab.

collected [45, 67] for learning visual navigation, leading to significant improvement in agent’s performance. However, the process towards such large-scale training involves solving a series of key sub-problems such as how to build navigation graphs [10, 30, 37], how to recover corrupted rendered images [8, 43], and how to generate navigational instructions [23, 25, 78, 83], which significantly influence the quality of collected data and should be investigated thoroughly. Meanwhile, an agent capable of understanding human natural language and navigating in photo-realistic environments is a complex and modularized system [3, 14, 18, 32, 76, 79, 90, 93], and it is important to study how to effectively utilize the large-scale data to benefit the training of navigational agents adequately.

In this paper, we propose an effective paradigm for large-scale vision-and-language navigation (VLN) training and quantitatively evaluate the influence of each component in the pipeline. Specifically, we utilize environments in both the HM3D [66] and the Gibson [84] datasets, build navigation graphs for the environments based on the Habitat simulator [70], sample new trajectories and generate corresponding instructions [74], and train agents [16, 18] for solving downstream navigation tasks [10, 35, 44, 60, 75]. Different from previous methods such as AutoVLN [17] and MARVAL [37], we build navigation graphs using an excessive viewpoint sampling and aggregation algorithm, following the graph construction heuristic proposed in [30], which results in fully-connected graphs with high coverage in open space. Additionally, we address the issue of corrupted rendered images from HM3D and Gibson environments with the Co-Modulated GAN [87], which we train to generate photo-realistic images from the faulty rendered images with broken, distorted, or missing regions, to mitigate the noise in visual data. Unlike MARVAL, which uses a non-public language generation model Marky [78] and visual encoder MURAL [34], as well as synthesizes observations from novel viewpoints with an image-to-image GAN [40], our large-scale training regime is fully reproducible and straightforward to execute, while leading to a significant improvement on agent’s performance.

Through comprehensive experiments, we find that a fully traversable navigation graph is crucial to improve the agent’s performance for downstream tasks with detailed instructions like R2R. Besides, we show that recovering photo-realistic images from the rendered images is very beneficial, especially for the low-quality 3D scans from the Gibson environments. Results also suggest that an agent can consistently benefit from having more diverse visual data, and learning from additional scenes helps agents to generalize better to unseen environments than simply learning from more data. Moreover, we validate that an agent trained with augmented instructions generated by a simple LSTM-based model [74] can achieve good performance on

multiple navigation tasks [10, 60, 75]. Last but not least, we find that appropriately combining our augmented data with the original data in pre-training and fine-tuning can benefit the agent’s generalization ability.

Remarkably, by following the above analysis as data augmentation and agent training guidelines, our resulting VLN model achieves 80% success rate (SR) on the R2R test split by simple imitation learning without pre-exploration [26, 74, 92], beam search [25, 54, 85] or model ensembling [63], and successfully eliminates the gap between navigating in seen and unseen environments. This result significantly outperforms previous best method (73%) [3], and reduces the difference towards human performance (86% SR³) [10] to 6%. Our method also achieves new state-of-the-art results on different language-guided visual navigation problems, including CVDN [75] and REVERIE [60]. Moreover, although the augmented data is discrete, it helps boost VLN performance in continuous environments (R2R-CE) [5, 30, 44], a much more realistic but difficult scenario, by 5% SR. All the results demonstrate the great effectiveness and generalization potential of our training regime. In summary, our main contributions include:

1. A simple, effective, fully automated and reproducible large-scale training paradigm for vision-and-language navigation.
2. Comprehensive analysis of the entire data augmentation pipeline and utilizing the large data for training.
3. New state-of-the-art results on navigation tasks including R2R, CVDN, REVERIE, and R2R-CE.

2. Related Works

Vision-and-Language Navigation Learning to navigate in unvisited environments following natural language instructions is an important step toward intelligent robots that can assist humans with daily activities. In the past years, a great variety of scenarios have been proposed for VLN research, such as navigation with comprehensive language guidance [10, 35, 45], navigation by interpreting dialog history [19, 56, 75], grounding remote objects with high-level instructions [60, 91], and navigation in continuous environments that closely approximate the real world [41, 44]. To address the problem, early research mainly focuses on developing task-specific models and training methods to better exploit visual-textual correspondence for decision making [2, 7, 21, 38, 48, 52, 58, 59, 77].

Large-Scale Visual Navigation Learning Due to the expensive navigational data collection process, learning to navigate usually faces a data scarcity issue [1, 10, 11, 12, 24, 45, 60, 75]. Many works have been proposed to scale up the

³Note that human followers only have egocentric views, while our model follows the common approach of applying panoramic observations.

training data by collecting more human annotations [67] or creating new environments [20, 66]. Moreover, recent studies tend to establish a scalable regime, utilizing extensive automatically-generated data to push the limit of agent performance [17, 37], or introducing large-scale pre-training approaches to improve the generalizing ability [16, 36, 62]. In this paper, we create a simple paradigm for scaling VLN training, and through comprehensive analysis, we seek a valuable guideline for data acquisition and agent training for future research.

3. Scaling Data for Learning VLN

We outline the necessary resources for learning VLN, followed by the details of our method for creating the large-scale augmented dataset from additional environments. Note that in this section, we only present our method to generate instruction-path pairs in R2R-style, which will be shared to address downstream R2R [10], CVDN [75] and R2R-CE [44] tasks. We refer to the *Appendix A* for data collection and model training details for REVERIE, whose data requires trajectories that lead to a specific object [60].

3.1. Resources for VLN Training

Most existing research on VLN is established over the discrete Matterport 3D environments (MP3D) [13] where an agent’s positions and observations are constrained on viewpoints of predefined navigation graphs. The trajectory-instruction pairs are sampled and annotated based on these discrete graphs. Compared to navigation in continuous environments [44, 70], such simplification enables efficient learning and execution while remaining to be practical, because, essentially, VLN agents make decisions by executing a vision-and-language grounding process [10]. There are also some recent works that attempt to transfer agents designed for discrete scenarios to continuous environments [9, 30, 42, 43]. Our data augmentation paradigm produces discrete supervisions, whereas we show in experiments that it also facilitates VLN learning in continuous scenes. In summary, scaling VLN data typically requires collecting new visual environments, discretizing the environments by building navigation graphs, sampling trajectories (sequences of images) on the graphs, and generating corresponding instructions. Following this procedure, we specify our data augmentation paradigm below.

3.2. Generating Augmented Data

Collecting Environments We adopt environments from HM3D [66] and Gibson [84] as the source of our visual data. Both datasets contain abundant, traversable, and simulated indoor 3D scans collected from real-world buildings, which support the learning of various visual navigation problems [12, 44, 55, 70]. Specifically, we employ 800

training scenes from HM3D, and 491 training and validation scenes from Gibson (same as MARVAL [37]), resulting in more than 150k m^2 navigable area, which is around $\times 7.5$ times larger than the training scenes of downstream MP3D environments (20k m^2 , 61 scans).

Constructing Navigation Graphs We argue that a high-quality navigation graph needs to satisfy a number of criteria, including high coverage of the space to maximize visual diversity and fully traversable edges in appropriate lengths with nodes positioned close to the center of open space for sampling reasonable trajectories. Previous work AutoVLN [17] builds graphs with very sparse nodes and with edges that go across obstacles, limiting the quantity of sampled data and leading to impractical trajectories, while MARVAL [37] trains a model to predict navigable directions, which could make errors and overcomplicate the problem. In this work, we propose a very simple but accurate heuristic for building the graph: we first apply the existing navigable position sampling function in Habitat simulator [70] to sample an excessive amount of viewpoints which almost covers the entire open space while limiting the geodesic distance between any two viewpoints to be greater than 0.4 m . Then, we apply the Agglomerative Clustering algorithm to group adjacent viewpoints to a single viewpoint with a distance threshold of 1.0 m , automatically producing positions close to the center of open space. We create a rough graph by randomly connecting viewpoints within 5.0 m separation while capping the maximal edges of a viewpoint to be five, and use the existing graph refinement approach [30] to obtain the final fully-connected and fully-traversable navigation graphs. We use this method to construct graphs for the 800+491 environments; the average edge length in the graphs is 1.41 m , and the average node degree is 4.55. We visualize the graphs in *Appendix C*.

Recovering Faulty Rendered Images Although HM3D and Gibson provide a large amount of diverse indoor environments, the quality of the images rendered from their 3D meshes is often much worse than the camera-captured images (as shown in Figure 2). Previous work has shown that navigation agents trained with rendered views will perform significantly worse than agents trained with high-quality images [43]. As a result, we consider recovery of faulty rendered images as a process in our ScaleVLN paradigm.

We formulate this task as an image-to-image translation problem, where the model takes a rendered image as input and learns to recover the broken, distorted, or missing regions. Specifically, we adopt the Co-Modulated GAN (Co-Mod GAN) [88], a generative model that can leverage conditional information and retain the stochastic in unconditional generation. We train Co-Mod GAN on the rendered-and-camera-image pairs in Matterport3D datasets and use

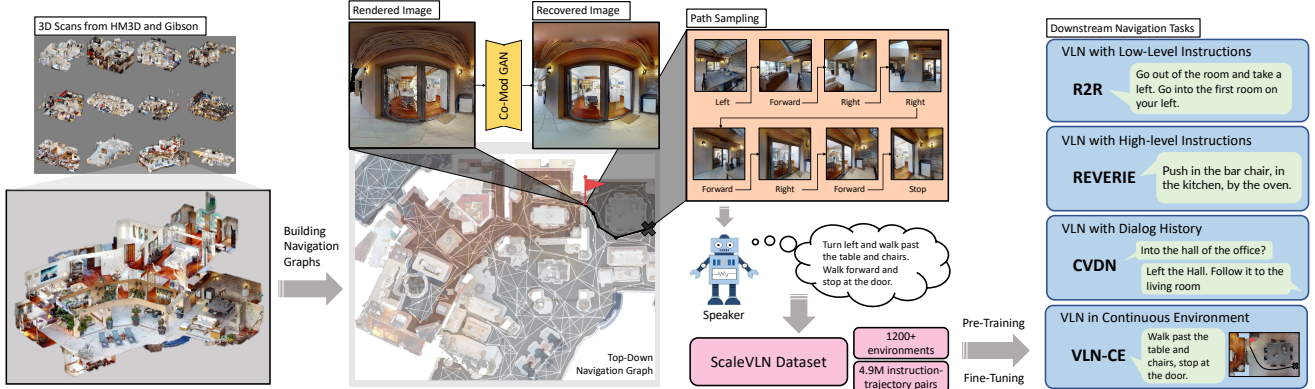


Figure 2: Our proposed paradigm (ScaleVLN) for generating large-scale augmented VLN data. ScaleVLN applies 1200+ unannotated 3D scans from the HM3D [66] and Gibson [84] environments, builds navigation graphs for each scene, recovers faulty rendered images with a Co-Mod GAN [87], samples trajectories and generates corresponding instructions, resulting in 4.9M augmented data to facilitate learning various downstream language-guided navigation tasks [10, 44, 60, 75].

the trained model to recover the rendered images in HM3D and Gibson environments.

Sample Trajectories We sample trajectories on the navigation graphs of HM3D and Gibson environments. For navigation tasks with detailed instructions, we follow PREVALENT [28] and collect all possible shortest routes between any two viewpoints connected by three to five intermediate nodes. This sampling strategy yields a total of 2,890,267 paths and 2,051,443 paths for the HM3D and Gibson environments, respectively.

Generate Navigational Instructions Finally, we apply the off-the-shelf EnvDrop Speaker [74], a simple LSTM-based language generation model trained on instruction-path pairs in R2R [10], to produce a navigational instruction for each sampled trajectory for navigation tasks with detailed instructions. Compared to the more powerful language model GPT-2 [65], EnvDrop Speaker generates less diverse descriptions, but the resulting data can lead to similar improvement on agents addressing R2R task (see §4.2).

Following the procedure above, our large-scale data augmentation paradigm creates 4,941,710 instruction-trajectory pairs for learning VLN. This size is $\times 352$ larger than the R2R dataset and $\times 4.62$ larger than the commonly applied augmented PREVALENT dataset [28].

4. Experiments

In this section, we present a comprehensive evaluation of the effect of each component in our data augmentation paradigm, investigate how to appropriately use the data for learning, and test agents [18, 16] pre-trained with our data on multiple VLN downstream tasks [10, 44, 60, 75].

4.1. Experimental Setup

Datasets We perform analysis mainly on the R2R dataset [10], while evaluating the generalization potential of our augmented data on REVERIE [60], CVDN [75] and R2R-CE [44]. The datasets are outlined as follows:

- **R2R** consists of 22k human-annotated navigational instructions, each describing a trajectory that traverses multiple rooms in MP3D [13]. On average, an instruction contains 32 words, and each ground-truth path is formed by seven nodes with a total length of 10 *m*.
- **REVERIE** inherits the trajectories in R2R but provides high-level instructions which describe a target object. The task for an agent is first to find the object, and localize it in observation.
- **CVDN** provides dialogues between a navigator who tries to find a target by asking for guidance and an oracle with a privileged view of the best next step. An agent who addresses the task needs to find the way by interpreting the dialogue history.
- **R2R-CE** transfers the discrete trajectories in R2R to continuous 3D scans rendered by Habitat simulator [70], where an agent can freely travel in the open space and need to interact with obstacles. The dataset contains 16k instruction-trajectory pairs after removing non-transferable paths.

Besides, we also adopt the widely applied augmented R2R dataset PREVALENT [28] in our experiments, which only has 178,270 samples created from MP3D scenes. For simplicity, we use PREV, HM-E, and Gib-E to denote PREVALENT data, our augmented data from HM3D and Gibson scenes with instructions generated by EnvDrop Speaker [74], respectively, and use the term ScaleVLN data for all our HM-E and Gib-E data in the following sections.

Baseline VLN Models We employ the recently proposed VLN agents, Dual-Scale Graph Transformer (DUET) [18] and History Aware Multimodal Transformer (HAMT) [16] as the baseline models in our experiments. The primary idea of DUET is to build a topological map on the fly, which extends the agent’s action space from its current viewpoint to all navigable directions encountered during navigation, therefore, greatly facilitating planning and error correction. HAMT explicitly stores the observations at each navigational step, which benefits the learning of sequence-to-sequence alignment between vision and instruction. We refer readers to their original papers for more technical details. In our experiments, we apply the DUET agent in R2R, CVDN and REVERIE, whereas using the HAMT agent in R2R-CE, since the two models report the best results on these datasets, respectively.

Training We use the augmented data for a two-stage VLN agent training, *i.e.*, pre-training and fine-tuning. In pre-training, we consider the most widely applied proxy tasks in previous work, Masked Language Modeling (MLM), Masked Region Modeling (MRM), and Single-Action Prediction (SAP) [16, 18, 28, 61, 62], to enhance agent’s language understanding, visual perception, and to benefit cross-modal grounding between instruction and observation (whose effect will be studied in §4.3). We refer to *Appendix A* for their implementation details.

After pre-training, similar to AutoVLN [18] and MARVAL [37], we fine-tune the model simply with imitation learning (IL) method DAGGER [69]. Specifically, at each time step, an agent performs an action sampled from the predicted probability of its action space, and minimizes the loss between the sampled action and the ground truth. This method allows an agent to learn from paths that cover wide space and reduces the exposure bias caused by teacher forcing [46].

Implementation Details We apply CLIP ViT-B/16 [64], a visual transformer [22] pre-trained to align millions of image-text pairs from the web, as the visual encoder in all our experiments if not specified otherwise. We refer to *Appendix A* for more details.

To address R2R and CVDN, we pre-train DUET for 20k iterations with a batch size of 256 and learning rate of 5×10^{-5} on two NVIDIA Tesla A100 GPUs for about 72 GPU hours. We select one of the models logged in pre-training for fine-tuning according to the accuracy in solving proxy tasks and the performance in following R2R instructions. The selected model is then fine-tuned for 200k iterations with batch size 16 on a single GPU on both R2R and ScaleVLN datasets, which takes about 48 GPU hours to reach the peak performance. For CVDN, we directly fine-tune the pre-trained DUET model on the dataset with

Methods	HM3D Nav Graphs		R2R Val-Seen			R2R Val-Unseen		
	Density	Collision	NE↓	SR↑	SPL↑	NE↓	SR↑	SPL↑
None	–	–	2.51	76.89	69.71	3.06	72.92	62.82
AutoVLN	0.36	29.35%	1.90	84.43	79.10	3.08	72.75	62.56
Ours	1.16	0.00%	2.25	79.82	75.06	2.75	76.01	66.94

Table 1: Comparison on navigation graphs. *Density* is computed as the number of nodes per navigable area (node/ m^2), and *Collision* is the ratio of edges that go through obstacles. *None* means the agent only learn from R2R and PREV data).

the same configurations as fine-tuning on R2R. For R2R-CE, we pre-train a single HAMT model with the same data and configurations, then fine-tune the model on the dataset. Similar to prior work [4, 5, 81], our HAMT agent in R2R-CE leverages a candidate waypoint predictor [30] which predicts navigable locations to support agent’s high-level decision-making process.

Evaluation Metrics Standard metrics [10] are applied to assess the agent’s performance, including Trajectory Length (TL) which is the average length of the agent’s predicted path in meters, Navigation Error (NE) which is the average distance between the agent’s final position and the target in meters, Success Rate (SR) which is the ratio of agents that stop within 3 meters to the target viewpoint, and Success penalized by Path Length (SPL) [6]. For R2R-CE, normalized Dynamic Time Warping (nDTW) is an additional metric that measures the step-wise alignment between the ground truths and the agent-predicted paths [33]. For CVDN, Goal Progress (GP) is the only metric; it measures the average difference between the length of the completed trajectory and the remaining distance to goal [75].

4.2. Scale VLN Data, What Really Matters?

Effect of Navigation Graphs We first study the effect of different navigation graphs in Table 1, where we compare the graphs from our method to AutoVLN [17]. For fairness, both methods only use 800 HM3D scenes without recovering faulty rendered images. We can see that generating augmented data from AutoVLN’s graph cannot benefit the agent’s performance in unseen environments. We suspect this is mainly due to a high ratio of edges that go through obstacles, resulting in noisy and misleading trajectories that do not exist in the downstream navigation graph. On the contrary, our fully traversable graphs with a high density of viewpoints produce effective data, which greatly improves the results, suggesting the importance of graph quality in sampling discrete augmented data.

Effect of More Data Table 2 shows the influence of the quantity of additional environments and training data. We can see that with the same amount of augmented scenes

HM-E Aug		R2R Val-Seen				R2R Val-Unseen			
#Scenes	#Samples	TL	NE↓	SR↑	SPL↑	TL	NE↓	SR↑	SPL↑
800	2890k	12.63	2.27	79.24	73.34	12.83	2.62	76.59	67.74
800	1400k	12.21	2.18	80.71	75.92	12.97	2.71	76.01	66.56
800	700k	12.63	1.86	83.35	77.97	13.83	2.69	76.25	66.00
400	700k	12.76	1.87	82.96	77.05	13.80	2.78	75.22	65.32
200	700k	11.95	1.95	82.66	77.97	13.29	2.73	74.84	64.59
0	0	13.28	2.51	76.89	69.71	13.53	3.06	72.92	62.82

Table 2: Comparison of the quantity of augmented scene and samples. Here each experiment is pre-trained on data from R2R, PREV, and HM-E, and fine-tuned on R2R.

Pre-Train	Fine-Tune	R2R Val-Seen			R2R Val-Unseen		
		NE↓	SR↑	SPL↑	NE↓	SR↑	SPL↑
R2R, PREV	R2R	2.51	76.89	69.71	3.06	72.92	62.82
R2R, PREV + HM-E	R2R	2.27	79.24	73.34	2.62	76.59	67.74
R2R, PREV + ScaleVLN	R2R	2.02	80.51	74.88	2.53	78.08	68.31
R2R, PREV	–	3.77	67.19	64.49	5.80	47.42	45.30
R2R, PREV + HM-E	–	4.04	64.64	62.00	5.03	55.09	52.23
R2R, PREV + ScaleVLN	–	3.64	71.11	68.53	4.90	57.00	54.03

Table 3: Results of adding more augmented data and the pre-trained model performance without fine-tuning.

(800), agent performance in val-unseen gradually increases with higher sampling density. On the other hand, generating the same amount of samples (700k) from more environments leads to better results. And it is clear that #Scenes has a stronger impact than #Samples, which suggests the importance of having more diverse environments for learning VLN. Then, in Table 3, we further increase the quantity of training samples from Gibson environments for comparison and evaluate the pre-trained model’s performance on R2R without fine-tuning. Conclusions from the previous table still hold: adding more scenes and data can bring a steady performance gain to the agent.

Effect of Image Quality We evaluate the influence of image quality in augmented data on agent performance in Table 4. We can see that for data from both HM3D and Gibson, recovering the rendered raw images can provide a noticeable benefit to the agent’s results. Such improvement is more apparent for Gib-E because a large portion of 3D meshes for reconstructing Gibson scenes is in low-quality [84], which leads to largely broken or distorted rendered views. In fact, HM-E (R) + Gib-E (F) leads to worse results than HM-E (R) alone even with 61% more different environments, suggesting the great importance of having data with high visual quality.

Effect of Augmented Instruction By comparing different speakers in Table 5, we found that a simple LSTM-based model (EnvDrop [74]) trained from-scratch results in a higher Bleu-4 score [57] than a fine-tuned GPT-2 [65]. However, we are aware that producing high-fidelity and detailed navigational instructions is a long-lasting and challenging problem [23, 25, 53, 74, 78, 90], but we only exper-

Scenes	R2R Val-Seen				R2R Val-Unseen			
	TL	NE↓	SR↑	SPL↑	TL	NE↓	SR↑	SPL↑
HM-E (F)	12.17	2.25	79.82	75.06	12.64	2.75	76.01	66.94
HM-E (R)	12.63	2.27	79.24	73.34	12.83	2.62	76.59	67.74
HM-E (R) + Gib-E (F)	12.74	2.17	81.00	75.44	13.57	2.65	76.33	66.97
HM-E (R) + Gib-E (R)	12.41	2.02	80.51	74.88	13.16	2.53	78.08	68.31

Table 4: Effect of augmented image quality. (F) denotes faulty rendered images and (R) denotes recovered images.

Speaker	Instruction Quality	R2R Val-Unseen			
	Bleu-4↑	TL	NE↓	SR↑	SPL↑
GPT-2	24.36	13.98	2.74	75.82	66.08
EnvDrop	27.66	12.83	2.62	76.59	67.74

Table 5: Quality of generated instructions and their influence on agent’s performance on the R2R dataset.

imented with two simple models. Numbers in Table 5 indicate that the generated instructions are of low quality while showing a large influence on learning to navigate, implying that pairing the augmented trajectories with better instructions could be promising future work.

4.3. How to Utilize Large-Scale Data?

Data for Pre-Training and Fine-Tuning Pre-training and fine-tuning are two essential stages where augmented data can directly impact. In Table 6, we investigate how to effectively apply the original R2R dataset, PREV [28], and our HM-E data in the two processes. First, comparing applying PREV and HM-E (Method#2 and #3) in pre-training, it is unsurprisingly that an agent benefits more from learning in environments different from downstream scenes. A better result of Method#4 shows that PREV and our HM-E complement each other in the pre-training phase. Then, we investigate the effect of applying augmented data in fine-tuning, in which the motivation is to avoid overfitting the small downstream dataset. Compare Method#4 to Method#5, #6, and #7; it is clear that it is very beneficial to keep the data augmented from the addition environments (HM-E) in fine-tuning (+2.51% SR in Val-Unseen). Moreover, by doing so, the generalization gap between navigating in seen and unseen environments has been reduced to less than 1% SR (80.02% vs. 79.10%), reflecting the importance of maintaining high visual diversity in training. Compare Method#6 and Method#7, including PREV in fine-tuning harms the performance likely because it will cause the learning to overfit the 61 MP3D scenes. In addition to Table 6, our experiment shows that adding Gib-E to the pre-training phrase can improve the result, while applying it in fine-tuning does not show a noticeable difference. This is likely because the generated instruction-trajectory pairs from Gibson environments have a larger gap to the Matterport scenes, which will introduce noise and is unsuitable for fine-tuning.

Method #	Pre-training Data			Fine-tuning Data			R2R Val-Seen				R2R Val-Unseen			
	R2R	PREV	HM-E (ours)	R2R	PREV	HM-E (ours)	TL	NE↓	SR↑	SPL↑	TL	NE↓	SR↑	SPL↑
1	✓			✓			12.02	3.39	69.83	64.30	12.45	4.04	65.26	56.91
2	✓	✓		✓			13.28	2.51	76.89	69.71	13.53	3.06	72.92	62.82
3	✓		✓	✓			12.80	2.69	75.02	68.71	12.66	2.79	74.96	65.90
4	✓	✓	✓	✓			12.63	2.27	79.24	73.34	12.83	2.62	76.59	67.74
5	✓	✓	✓	✓	✓		12.31	2.20	80.51	75.75	12.86	2.65	75.78	66.36
6	✓	✓	✓	✓		✓	13.38	2.12	80.02	73.52	13.32	2.46	79.10	68.66
7	✓	✓	✓	✓	✓	✓	12.62	2.18	80.71	75.29	13.22	2.58	77.10	67.23

Table 6: Influence of applying augmented data in pre-training and fine-tuning on agent’s performance.

Pre-training Tasks			R2R Val-Seen				R2R Val-Unseen			
MLM	SAP	MRM	TL	NE↓	SR↑	SPL↑	TL	NE↓	SR↑	SPL↑
✓			11.96	3.61	66.01	60.66	14.47	4.26	62.62	51.25
			14.13	2.91	74.93	65.95	15.75	3.62	69.14	57.47
	✓		12.20	1.99	80.61	75.07	12.89	2.89	74.20	65.40
✓	✓		12.63	2.27	79.24	73.34	12.83	2.62	76.59	67.74
✓	✓	✓	12.37	1.97	81.88	75.72	13.69	2.73	75.82	66.62

Table 7: Influence of pre-training tasks. MLM, SAP, and MRM denote masked language modeling, single-action prediction, and masked region modeling.

Methods	Val-Unseen			Test-Unseen		
	OSR↑	SR↑	SPL↑	OSR↑	SR↑	SPL↑
RecBERT [31]	27.66	25.53	21.06	26.67	24.62	19.48
SIA [49]	44.67	31.53	16.28	44.56	30.80	14.85
HAMT [16]	36.84	32.95	30.20	33.41	30.40	26.67
DUET [18]	51.07	46.98	33.73	56.91	52.51	36.06
AutoVLN [17]	62.14	55.89	40.85	62.30	55.17	38.88
DUET+ScaleVLN (ours)	63.85	56.97	41.84	62.65	56.13	39.52

Table 8: Navigation performance on REVERIE dataset.

Effect of Pre-training Tasks In Table 7, we further investigate the effect of three proxy tasks, MLM, SAP, and MRM, on pre-training the best performing model in Table 6 (Method#6). Results show that both MLM and SAP are very effective pre-training tasks that can greatly enhance the agent’s performance when applied alone, and they are complementary since combining the two tasks can lead to a larger improvement (+16.49% SPL higher than without pre-training). However, learning MRM with the other two proxy tasks slightly degenerates the results. We suspect this is because an agent can already learn very rich and generalizable semantic representations from large and diverse augmented data, whereas predicting the probability distribution of object categories for masked images introduces too much noise to the learning process.

Based on the findings from our experiments, we pre-train our agent with MLM and SAP on R2R, PREV, and our ScaleVLN datasets to get the best pre-trained model, and fine-tune on R2R and HM-E for best performance.

4.4. Evaluate on Various VLN Tasks

R2R Table 9 compares agents’ single-run performance on the R2R dataset. We can see that training DUET model with our ScaleVLN data results in 8% SR and 8% SPL absolute improvement on the test split⁴, which also greatly outperforms the previous best method BEVbert [3]. As suggested in MARVAL [37], we also experiment with applying a more powerful visual encoder, CLIP ViT-H/14 [64], and the image augmentation method EnvEdit [47] to our approach, leading to a remarkable 80% SR, and reducing the long-lasting generalization gap between seen and unseen environments [86] to less than 1%. It is interesting to notice that the remaining gap towards human performance (6% SR) is similar to the difference between the agent’s OSR and SR (6~7%), which suggests that it might be important for future work to improve the policy network to tackle the stopping problem given large-scale data.

REVERIE We show in Table 8 that our method achieves the new state-of-the-art results in all metrics on the REVERIE task. Our method surpasses AutoVLN, which uses all the 1000 HM3D environments for pre-training, by 0.94% in success rate and 0.64% in SPL on the test leaderboard with only 800 HM3D scenes and 491 Gibson low-quality environments. This again validates the effectiveness of our high-quality connectivity graphs and image recovery in our large-scale training paradigm.

CVDN As shown in Table 10, our method achieves the new state-of-the-art performance on the CVDN test-unseen split, which largely improves the goal progress (GP) of the previous SoTA by 1.41 meters (a relative gain of 25.26%). This result shows that our R2R-style augmented data can generalize to a different VLN task with a distinct type of instructions, likely because visual scarcity is the major bottleneck in learning VLN, as suggested in Table 2.

⁴On the R2R test-unseen leaderboard: <https://eval.ai/web/challenges/challenge-page/97/leaderboard/270>, our method surpasses all single-run results and outperforms all previous models applying beam-search or pre-exploration (see Appendix B).

Methods	R2R Val-Seen					R2R Val-Unseen					R2R Test-Unseen				
	TL	NE↓	OSR↑	SR↑	SPL↑	TL	NE↓	OSR↑	SR↑	SPL↑	TL	NE↓	OSR↑	SR↑	SPL↑
Human	–	–	–	–	–	–	–	–	–	–	11.85	1.61	90	86	76
Seq2Seq [10]	11.33	6.01	53	39	–	8.39	7.81	28	21	–	8.13	7.85	27	20	–
Speaker Follower [25]	–	3.36	74	66	–	–	6.62	45	36	–	14.82	6.62	–	35	28
RCM [79]	10.65	3.53	75	67	–	11.46	6.09	50	43	–	11.97	6.12	50	43	38
SSM [76]	14.70	3.10	80	71	62	20.70	4.32	73	62	45	20.40	4.57	70	61	46
EnvDrop [74]	11.00	3.99	–	62	59	10.70	5.22	–	52	48	11.66	5.23	59	51	47
PREVALENT [28] †	10.32	3.67	–	69	65	10.19	4.71	–	58	53	10.51	5.30	61	54	51
EntityGraph [29]	10.13	3.47	–	67	65	9.99	4.73	–	57	53	10.29	4.75	61	55	52
NvEM [2]	11.09	3.44	–	69	65	11.83	4.27	–	60	55	12.98	4.37	66	58	54
AirBert [27] ††	11.09	2.68	–	75	70	11.78	4.10	–	62	56	12.41	4.13	–	62	57
VLN⊙BERT [31] †	11.13	2.90	–	72	68	12.01	3.93	–	63	57	12.35	4.09	70	63	57
MARVAL [37] ††	10.60	2.99	–	73	69	10.15	4.06	–	65	61	10.22	4.18	67	62	58
EnvMix [51] †	10.88	2.48	–	75	72	12.44	3.89	–	64	58	13.11	3.87	72	65	59
HAMT [16] †	11.15	2.51	–	76	72	11.46	2.29	–	66	61	12.27	3.93	72	65	60
SnapEnsemble [63] †°	–	–	–	–	–	12.05	3.63	–	67	60	12.71	3.82	–	65	60
HOP+ [62] †	11.31	2.33	–	78	73	11.76	3.49	–	67	61	12.67	3.71	–	66	60
TD-STP [89] †	–	2.34	83	77	73	–	3.22	76	70	63	–	3.73	72	67	61
DUET [18] †	12.32	2.28	86	79	73	13.94	3.31	81	72	60	14.73	3.65	76	69	59
BEVBert [3] †	13.56	2.17	88	81	74	14.55	2.81	84	75	64	15.87	3.13	81	73	62
DUET+ScaleVLN (ours) ††	11.90	2.16	87	80	75	12.40	2.34	87	79	70	14.27	2.73	83	77	68
DUET*+ScaleVLN (ours) ††	13.24	2.12	87	81	75	14.09	2.09	88	81	70	13.93	2.27	86	80	70

Table 9: Comparison of single-run performance on R2R dataset. †: Methods that apply vision-language-action pre-training. ††: Methods that use additional visual data than MP3D. °: Model ensemble. *: Applying EnvEdit as image augmentation and CLIP ViT-H14 as image features.

Methods	Val-Seen	Val-Unseen	Test-Unseen
	GP↑	GP↑	GP↑
PREVALENT [28]	–	3.15	2.44
MT-RCM+EnvAg [80]	5.07	4.65	3.91
NDH-Full [39]	–	5.51	5.27
HAMT [16]	6.91	5.13	5.58
MTVM [48]	–	5.15	4.82
DUET+ScaleVLN (Ours)	8.13	6.12	6.97

Table 10: Navigation performance on CVDN dataset.

Methods	R2R-CE Val-Unseen				R2R-CE Test-Unseen		
	NE↓	nDTW↑	SR↑	SPL↑	NE↓	SR↑	SPL↑
CMA [44]	7.37	40	32	30	7.91	28	25
LAW [68]	–	–	35	31	–	–	–
Waypoint Models [42]	6.31	–	36	34	6.65	32	30
WS-MGMap [15]	6.28	–	39	34	7.11	35	28
VLN⊙BERT† [30]	5.74	53	44	39	5.89	42	36
Sim2Sim [43]	6.07	–	43	36	6.17	44	37
VLN⊙BERT+Ego ² -Map† [32]	4.94	60	52	46	5.54	47	41
HAMT+InternVideo† [81]	4.95	62	53	48	–	–	–
HAMT+ScaleVLN† (ours)	4.80	64	55	51	5.11	55	50

Table 11: Navigation performance on the R2R-CE datasets. †: Methods that applies candidate waypoint predictor [30] to support high-level action space.

R2R-CE Although our augmented ScaleVLN data only contains discrete instruction-trajectory pairs, it can benefit the agent’s performance in continuous environments with the support of the candidate waypoint predictor [30] (Table 11). Compared to the previous method, which applies very strong pre-trained visual representations [32, 81], our method still demonstrates obvious improvement, which reflects the effectiveness of generating in-domain data for learning VLN.

5. Conclusion

In this paper, we introduce a simple but effective large-scale data generation paradigm for learning vision-and-language navigation (ScaleVLN). The method applies thousands of photo-realistic environments from HM3D and Gibson datasets, and creates millions of instruction-trajectory pairs for training. Apart from the unsurprising improvement of learning from abundant visual data in agent performance, we demonstrate the effectiveness of building high-quality navigation graphs and using camera-quality images through comprehensive experiments. Moreover, we investigate how to properly utilize the augmented data in pre-training and fine-tuning an agent, as well as the influence of different pre-training tasks on the downstream navigation results. By following our findings as data augmentation and agent training guidelines, we achieve new state-of-the-art results on several VLN benchmarking datasets that cover distinct styles of instructions (R2R, REVERIE, CVDN) and action spaces (R2R-CE). We believe our ScaleVLN paradigm can be easily applied as a tool to facilitate data augmentation for VLN and other visual navigation problems, and the experiments presented in the paper can provide useful insights for future research in creating and utilizing large-scale data.

6. Acknowledgement

We thank ICCV reviewers for their helpful suggestions. This work is partially supported by the National Key R&D Program of China (NO. 2022ZD0160100), and in part by the Shanghai Committee of Science and Technology (Grant No. 21DZ1100100). We warmly thank Taesung Park for helpful suggestions on recovering rendered images.

References

- [1] Abhishek Kadian, Joanne Truong, Aaron Gokaslan, Alexander Clegg, Erik Wijmans, Stefan Lee, Manolis Savva, Sonia Chernova, and Dhruv Batra. Are We Making Real Progress in Simulated Environments? Measuring the Sim2Real Gap in Embodied Visual Navigation. In *arXiv:1912.06321*, 2019. 2
- [2] Dong An, Yuankai Qi, Yan Huang, Qi Wu, Liang Wang, and Tieniu Tan. Neighbor-view enhanced model for vision and language navigation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 5101–5109, 2021. 2, 8
- [3] Dong An, Yuankai Qi, Yangguang Li, Yan Huang, Liang Wang, Tieniu Tan, and Jing Shao. Bevbnet: Topo-metric map pre-training for language-guided navigation. *arXiv preprint arXiv:2212.04385*, 2022. 2, 7, 8
- [4] Dong An, Hanqing Wang, Wenguan Wang, Zun Wang, Yan Huang, Keji He, and Liang Wang. Etpnav: Evolving topological planning for vision-language navigation in continuous environments. *arXiv preprint arXiv:2304.03047*, 2023. 5
- [5] Dong An, Zun Wang, Yangguang Li, Yi Wang, Yicong Hong, Yan Huang, Liang Wang, and Jing Shao. 1st place solutions for rxr-habitat vision-and-language navigation competition (cvpr 2022). *arXiv preprint arXiv:2206.11610*, 2022. 2, 5, 14
- [6] Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018. 5
- [7] Peter Anderson, Ayush Shrivastava, Devi Parikh, Dhruv Batra, and Stefan Lee. Chasing ghosts: Instruction following as bayesian state tracking. *Advances in neural information processing systems*, 32, 2019. 2
- [8] Peter Anderson, Ayush Shrivastava, Joanne Truong, Arjun Majumdar, Devi Parikh, Dhruv Batra, and Stefan Lee. Sim-to-real transfer for vision-and-language navigation. In *Conference on Robot Learning*, pages 671–681. PMLR, 2021. 2
- [9] Peter Anderson, Ayush Shrivastava, Joanne Truong, Arjun Majumdar, Devi Parikh, Dhruv Batra, and Stefan Lee. Sim-to-real transfer for vision-and-language navigation. In *Conference on Robot Learning*, pages 671–681. PMLR, 2021. 3
- [10] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3674–3683, 2018. 1, 2, 3, 4, 5, 8, 14, 15
- [11] Dhruv Batra, Angel X Chang, Sonia Chernova, Andrew J Davison, Jia Deng, Vladlen Koltun, Sergey Levine, Jitendra Malik, Igor Mordatch, Roozbeh Mottaghi, et al. Rearrangement: A challenge for embodied ai. *arXiv preprint arXiv:2011.01975*, 2020. 2
- [12] Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. ObjectNav Revisited: On Evaluation of Embodied Agents Navigating to Objects. In *arXiv:2006.13171*, 2020. 2, 3
- [13] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niebner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. In *2017 International Conference on 3D Vision (3DV)*, pages 667–676. IEEE, 2017. 3, 4
- [14] Jinyu Chen, Chen Gao, Erli Meng, Qiong Zhang, and Si Liu. Reinforced structured state-evolution for vision-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15450–15459, 2022. 2
- [15] Peihao Chen, Dongyu Ji, Kunyang Lin, Runhao Zeng, Thomas H Li, Mingkui Tan, and Chuang Gan. Weakly-supervised multi-granularity map learning for vision-and-language navigation. *arXiv preprint arXiv:2210.07506*, 2022. 8
- [16] Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. History aware multimodal transformer for vision-and-language navigation. *Advances in Neural Information Processing Systems*, 34:5834–5847, 2021. 2, 3, 4, 5, 7, 8, 13, 14, 15
- [17] Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. Learning from unlabeled 3d environments for vision-and-language navigation. In *European Conference on Computer Vision*, pages 638–655. Springer, 2022. 1, 2, 3, 5, 7, 13, 15, 16
- [18] Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. Think global, act local: Dual-scale graph transformer for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16537–16547, 2022. 2, 4, 5, 7, 8, 13, 15
- [19] Harm De Vries, Kurt Shuster, Dhruv Batra, Devi Parikh, Jason Weston, and Douwe Kiela. Talk the walk: Navigating new york city through grounded dialogue. *arXiv preprint arXiv:1807.03367*, 2018. 2
- [20] Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Jordi Salvador, Kiana Ehsani, Winson Han, Eric Kolve, Ali Farhadi, Aniruddha Kembhavi, et al. Proctor: Large-scale embodied ai using procedural generation. *arXiv preprint arXiv:2206.06994*, 2022. 1, 3
- [21] Zhiwei Deng, Karthik Narasimhan, and Olga Russakovsky. Evolving graphical planner: Contextual global planning for vision-and-language navigation. *Advances in Neural Information Processing Systems*, 33:20660–20672, 2020. 2
- [22] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 5, 13

- [23] Zi-Yi Dou and Nanyun Peng. Foam: A follower-aware speaker model for vision-and-language navigation. *arXiv preprint arXiv:2206.04294*, 2022. 2, 6
- [24] Kiana Ehsani, Winson Han, Alvaro Herrasti, Eli VanderBilt, Luca Weihs, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. Manipulathor: A framework for visual object manipulation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4497–4506, 2021. 2
- [25] Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. Speaker-follower models for vision-and-language navigation. In *Advances in Neural Information Processing Systems*, pages 3314–3325, 2018. 2, 6, 8
- [26] Tsu-Jui Fu, Xin Eric Wang, Matthew F Peterson, Scott T Grafton, Miguel P Eckstein, and William Yang Wang. Counterfactual vision-and-language navigation via adversarial path sampler. In *European Conference on Computer Vision*, pages 71–86. Springer, 2020. 2
- [27] Pierre-Louis Guhur, Makarand Tapaswi, Shizhe Chen, Ivan Laptev, and Cordelia Schmid. Airbert: In-domain pretraining for vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1634–1643, 2021. 1, 8
- [28] Weituo Hao, Chunyuan Li, Xiujun Li, Lawrence Carin, and Jianfeng Gao. Towards learning a generic agent for vision-and-language navigation via pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13137–13146, 2020. 4, 5, 6, 8, 14
- [29] Yicong Hong, Cristian Rodriguez, Yuankai Qi, Qi Wu, and Stephen Gould. Language and visual entity relationship graph for agent navigation. *Advances in Neural Information Processing Systems*, 33, 2020. 8
- [30] Yicong Hong, Zun Wang, Qi Wu, and Stephen Gould. Bridging the gap between learning in discrete and continuous environments for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15439–15449, 2022. 2, 3, 5, 8, 14
- [31] Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. A recurrent vision-and-language bert for navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1643–1653, June 2021. 1, 7, 8
- [32] Yicong Hong, Yang Zhou, Ruiyi Zhang, Franck Dernoncourt, Trung Bui, Stephen Gould, and Hao Tan. Learning navigational visual representations with semantic map supervision. *arXiv preprint arXiv: 2307.12335*, 2023. 2, 8
- [33] Gabriel Ilharco, Vihan Jain, Alexander Ku, Eugene Ie, and Jason Baldridge. General evaluation for instruction conditioned navigation using dynamic time warping. *arXiv preprint arXiv:1907.05446*, 2019. 5, 14
- [34] Aashi Jain, Mandy Guo, Krishna Srinivasan, Ting Chen, Sneha Kudugunta, Chao Jia, Yinfei Yang, and Jason Baldridge. Mural: Multimodal, multitask representations across languages. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3449–3463, 2021. 2
- [35] Vihan Jain, Gabriel Magalhaes, Alexander Ku, Ashish Vaswani, Eugene Ie, and Jason Baldridge. Stay on the path: Instruction fidelity in vision-and-language navigation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1862–1872, 2019. 2, 14
- [36] Mohit Bansal Jialu Li. Improving vision-and-language navigation by generating future-view image semantics. In *CVPR*, 2023. 3
- [37] Aishwarya Kamath, Peter Anderson, Su Wang, Jing Yu Koh, Alexander Ku, Austin Waters, Yinfei Yang, Jason Baldridge, and Zarana Parekh. A new path: Scaling vision-and-language navigation with synthetic instructions and imitation learning. *arXiv preprint arXiv:2210.03112*, 2022. 1, 2, 3, 5, 7, 8
- [38] Liyiming Ke, Xiujun Li, Yonatan Bisk, Ari Holtzman, Zhe Gan, Jingjing Liu, Jianfeng Gao, Yejin Choi, and Siddhartha Srinivasa. Tactical rewind: Self-correction via backtracking in vision-and-language navigation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6741–6749, 2019. 2
- [39] Hyounghun Kim, Jialu Li, and Mohit Bansal. Ndh-full: Learning and evaluating navigational agents on full-length dialogue. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021. 8
- [40] Jing Yu Koh, Harsh Agrawal, Dhruv Batra, Richard Tucker, Austin Waters, Honglak Lee, Yinfei Yang, Jason Baldridge, and Peter Anderson. Simple and effective synthesis of indoor 3d scenes. *arXiv preprint arXiv:2204.02960*, 2022. 2
- [41] Jacob Krantz, Shurjo Banerjee, Wang Zhu, Jason Corso, Peter Anderson, Stefan Lee, and Jesse Thomason. Iterative vision-and-language navigation. *arXiv preprint arXiv:2210.03087*, 2022. 2
- [42] Jacob Krantz, Aaron Gokaslan, Dhruv Batra, Stefan Lee, and Oleksandr Maksymets. Waypoint models for instruction-guided navigation in continuous environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15162–15171, 2021. 3, 8, 14
- [43] Jacob Krantz and Stefan Lee. Sim-2-sim transfer for vision-and-language navigation in continuous environments. In *European Conference on Computer Vision*, pages 588–603. Springer, 2022. 2, 3, 8
- [44] Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *European Conference on Computer Vision*, 2020. 2, 3, 4, 8, 14
- [45] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4392–4412, 2020. 2
- [46] Alex M Lamb, Anirudh Goyal ALIAS PARTH GOYAL, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio. Professor forcing: A new algorithm for

training recurrent networks. *Advances in neural information processing systems*, 29, 2016. 5

- [47] Jialu Li, Hao Tan, and Mohit Bansal. Envedit: Environment editing for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15407–15417, 2022. 1, 7
- [48] Chuang Lin, Yi Jiang, Jianfei Cai, Lizhen Qu, Gholamreza Haffari, and Zehuan Yuan. Multimodal transformer with variable-length memory for vision-and-language navigation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVI*, pages 380–397. Springer, 2022. 2, 8
- [49] Xiangru Lin, Guanbin Li, and Yizhou Yu. Scene-intuitive agent for remote embodied visual grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7036–7045, 2021. 7
- [50] Xiangru Lin, Guanbin Li, and Yizhou Yu. Scene-intuitive agent for remote embodied visual grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7036–7045, 2021. 13, 15
- [51] Chong Liu, Fengda Zhu, Xiaojun Chang, Xiaodan Liang, Zongyuan Ge, and Yi-Dong Shen. Vision-language navigation with random environmental mixup. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1644–1654, 2021. 8
- [52] Chih-Yao Ma, Jiasen Lu, Zuxuan Wu, Ghassan AlRegib, Zsolt Kira, Richard Socher, and Caiming Xiong. Self-monitoring navigation agent via auxiliary progress estimation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. 2
- [53] Aly Magassouba, Komei Sugiura, and Hisashi Kawai. Crossmap transformer: A crossmodal masked path transformer using double back-translation for vision-and-language navigation. *IEEE Robotics and Automation Letters*, 6(4):6258–6265, 2021. 6
- [54] Arjun Majumdar, Ayush Shrivastava, Stefan Lee, Peter Anderson, Devi Parikh, and Dhruv Batra. Improving vision-and-language navigation with image-text pairs from the web. In *Proceedings of the European Conference on Computer Vision*, 2020. 1, 2
- [55] Lina Mezghani, Sainbayar Sukhbaatar, Thibaut Lavril, Oleksandr Maksymets, Dhruv Batra, Piotr Bojanowski, and Kar-teek Alahari. Memory-augmented reinforcement learning for image-goal navigation. *arXiv preprint arXiv:2101.05181*, 2021. 3
- [56] Aishwarya Padmakumar, Jesse Thomason, Ayush Shrivastava, Patrick Lange, Anjali Narayan-Chen, Spandana Gella, Robinson Piramuthu, Gokhan Tur, and Dilek Hakkani-Tur. Teach: Task-driven embodied agents that chat. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2017–2025, 2022. 2
- [57] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002. 6
- [58] Yuankai Qi, Zizheng Pan, Yicong Hong, Ming-Hsuan Yang, Anton van den Hengel, and Qi Wu. The road to know-where: An object-and-room informed sequential bert for indoor vision-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1655–1664, 2021. 2
- [59] Yuankai Qi, Zizheng Pan, Shengping Zhang, Anton van den Hengel, and Qi Wu. Object-and-action aware model for visual language navigation. In *European Conference on Computer Vision*, 2020. 2
- [60] Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. Reverie: Remote embodied visual referring expression in real indoor environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9982–9991, 2020. 2, 3, 4
- [61] Yanyuan Qiao, Yuankai Qi, Yicong Hong, Zheng Yu, Peng Wang, and Qi Wu. Hop: History-and-order aware pre-training for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15418–15427, 2022. 5
- [62] Yanyuan Qiao, Yuankai Qi, Yicong Hong, Zheng Yu, Peng Wang, and Qi Wu. Hop+: History-enhanced and order-aware pre-training for vision-and-language navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 3, 5, 8
- [63] Wenda Qin, Teruhisa Misu, and Derry Wijaya. Explore the potential performance of vision-and-language navigation model: a snapshot ensemble method. *arXiv preprint arXiv:2111.14267*, 2021. 2, 8
- [64] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 5, 7, 13
- [65] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 4, 6
- [66] Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, et al. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. 1, 2, 3, 4
- [67] Ram Ramrakhya, Eric Undersander, Dhruv Batra, and Abhishek Das. Habitat-web: Learning embodied object-search strategies from human demonstrations at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5173–5183, 2022. 2, 3
- [68] Sonia Raychaudhuri, Saim Wani, Shivansh Patel, Unnat Jain, and Angel X Chang. Language-aligned waypoint (law) supervision for vision-and-language navigation in continuous environments. *arXiv preprint arXiv:2109.15207*, 2021. 8
- [69] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-

- regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011. 5
- [70] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9339–9347, 2019. 2, 3, 4
- [71] Dhruv Shah, Blazej Osinski, Brian Ichter, and Sergey Levine. Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action. *arXiv preprint arXiv:2207.04429*, 2022. 1
- [72] Sheng Shen, Liunan Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. How much can clip benefit vision-and-language tasks? In *International Conference on Learning Representations*, 2021. 1
- [73] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019. 14
- [74] Hao Tan, Licheng Yu, and Mohit Bansal. Learning to navigate unseen environments: Back translation with environmental dropout. In *Proceedings of NAACL-HLT*, pages 2610–2621, 2019. 1, 2, 4, 6, 8, 14
- [75] Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. Vision-and-dialog navigation. In *Conference on Robot Learning*, pages 394–406, 2020. 2, 3, 4, 5
- [76] Hanqing Wang, Wenguan Wang, Wei Liang, Caiming Xiong, and Jianbing Shen. Structured scene memory for vision-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8455–8464, 2021. 2, 8
- [77] Hanqing Wang, Wenguan Wang, Tianmin Shu, Wei Liang, and Jianbing Shen. Active visual information gathering for vision-language navigation. In *European Conference on Computer Vision*, pages 307–322. Springer, 2020. 2
- [78] Su Wang, Ceslee Montgomery, Jordi Orbay, Vighnesh Birodkar, Aleksandra Faust, Izzeddin Gur, Natasha Jaques, Austin Waters, Jason Baldridge, and Peter Anderson. Less is more: Generating grounded navigation instructions from landmarks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15428–15438, 2022. 2, 6
- [79] Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6629–6638, 2019. 2, 8
- [80] Xin Eric Wang, Vihan Jain, Eugene Ie, William Yang Wang, Zornitsa Kozareva, and Sujith Ravi. Environment-agnostic multitask learning for natural language grounded navigation. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*, pages 413–430. Springer, 2020. 8
- [81] Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, Sen Xing, Guo Chen, Juntong Pan, Jiashuo Yu, Yali Wang, Limin Wang, and Yu Qiao. Internvideo: General video foundation models via generative and discriminative learning. *arXiv preprint arXiv:2212.03191*, 2022. 5, 8
- [82] Erik Wijmans, Abhishek Kadian, Ari Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra. DD-PPO: Learning near-perfect pointgoal navigators from 2.5 billion frames. In *International Conference on Learning Representations (ICLR)*, 2020. 14
- [83] Zongkai Wu, Zihan Liu, Ting Wang, and Donglin Wang. Improved speaker and navigator for vision-and-language navigation. *IEEE MultiMedia*, 28(4):55–63, 2021. 2
- [84] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9068–9079, 2018. 1, 2, 3, 4, 6
- [85] Liang Xie, Meishan Zhang, You Li, Wei Qin, Ye Yan, and Erwei Yin. Vision–language navigation with beam-constrained global normalization. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. 2
- [86] Yubo Zhang, Hao Tan, and Mohit Bansal. Diagnosing the environment bias in vision-and-language navigation. *IJCAI*, 2020. 7
- [87] Shengyu Zhao, Jonathan Cui, Yilun Sheng, Yue Dong, Xiao Liang, Eric I Chang, and Yan Xu. Large scale image completion via co-modulated generative adversarial networks. *arXiv preprint arXiv:2103.10428*, 2021. 2, 4, 13, 15
- [88] Shengyu Zhao, Jonathan Cui, Yilun Sheng, Yue Dong, Xiao Liang, Eric I Chang, and Yan Xu. Large scale image completion via co-modulated generative adversarial networks. *arXiv preprint arXiv:2103.10428*, 2021. 3
- [89] Yusheng Zhao, Jinyu Chen, Chen Gao, Wenguan Wang, Lirong Yang, Haibing Ren, Huaxia Xia, and Si Liu. Target-driven structured transformer planner for vision-language navigation. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4194–4203, 2022. 8
- [90] Gengze Zhou, Yicong Hong, and Qi Wu. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. *arXiv preprint arXiv:2305.16986*, 2023. 2, 6
- [91] Fengda Zhu, Xiwen Liang, Yi Zhu, Qizhi Yu, Xiaojun Chang, and Xiaodan Liang. Soon: Scenario oriented object navigation with graph-based exploration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12689–12699, 2021. 2
- [92] Fengda Zhu, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. Vision-language navigation with self-supervised auxiliary reasoning tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10012–10022, 2020. 2
- [93] Yi Zhu, Yue Weng, Fengda Zhu, Xiaodan Liang, Qixiang Ye, Yutong Lu, and Jianbin Jiao. Self-motivated communication agent for real-world vision-dialog navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1594–1603, 2021. 2

Appendices

We first describe the implementation details of our experiments in Sec. A, including pre-training objectives and details of REVERIE experiments. In Sec. B, we provide additional experiments about the effects of visual encoders, model initialization, and adding depth features. We then discuss the impact of ScaleVLN on different VLN agents and on learning the long-horizon VLN task (R4R). Leaderboard results of R2R and object grounding results for REVERIE are also included. Sec. C and Sec. D visualize our navigability graphs and the recovered images from Co-Modulated GAN [87].

A. Implementation Details (§4⁵)

A.1. Pre-Training Objectives (§4.1)

We mainly employ three proxy tasks, MLM, MRM, and SAP, for pre-training the agent. Here we describe these proxy tasks in detail. The inputs for these tasks are instruction \mathcal{W} and demonstration path \mathcal{P} . During training, we randomly sample one task for each iteration with equal probability.

Masked Language Modeling (MLM) involves predicting masked words based on textual context and the full trajectory. A special [mask] token is used to randomly mask out 15% of the tokens in \mathcal{W} . We predict the masked word distribution $p(w_i|\mathcal{W}_{\setminus i}, \mathcal{P}) = f_{\text{MLM}}(x'_i)$ through a two-layer fully-connected network, where $\mathcal{W}_{\setminus i}$ is the masked instruction and x'_i is the output embedding of the masked word w_i . The objective is to minimize the negative log-likelihood of predicting the original words: $\mathcal{L}_{\text{MLM}} = -\log p(w_i|\mathcal{W}_{\setminus i}, \mathcal{P})$.

Masked Region Modeling (MRM) is to predict labels for masked regions in history observations based on instructions and neighboring regions. To achieve this, we randomly remove view images in \mathcal{P} with a 15% probability. For view images, the target labels are determined by an image classification model [22] pre-trained on ImageNet. To predict semantic labels for each masked visual token, we use a two-layer fully-connected network. The objective is to minimize the KL-divergence between the predicted and target probability distribution.

Single Action Prediction (SAP) aims to predict the next action based on the instruction and the given path. Following [18], we predict the probability for each candidate action in the action space via a two-layer fully-connected network. The objective is to minimize the negative log probability of the target view action $\mathcal{L}_{\text{SAP}} = -\log p_t(a_t^*|\mathcal{W}, \mathcal{P}_{<t})$.

⁵Link to Section 4 in Main Paper.

A.2. Implementation Details of REVERIE (§4.1)

REVERIE data contains trajectories that lead to target objects specified by high-level instructions. Following AutoVLN [17], for every visible object at a viewpoint, we sample paths with an edge length between 4 and 9 that end at the viewpoint. We filter out objects that are more than 3 meters away from the central of the viewpoint, resulting in 518,233 paths from HM3D, and 311,976 paths from the Gibson environments. To generate instructions in REVERIE-style, we modify the GPT-2 architecture used in AutoVLN [17] by only encoding the target object in the final viewpoint as the prompt to generate the instructions. Our large-scale data augmentation paradigm creates 830,209 instruction-trajectory pairs for training. This size is $\times 38$ larger than the original REVERIE dataset, and $\times 3.81$ larger than the augmented dataset in AutoVLN [17].

We follow DUET and SIA [50] to pre-train the model with an additional Object Grounding (OG) task, which requires selecting a target from object candidates based on high-level instruction and observations along the path. We use CLIP ViT-H/14 [64] to extract the image features, and ViT-B/16 [22] pre-trained on ImageNet to extract the object features. We pre-train DUET for 100k iterations with a batch size of 128 and a learning rate of 5×10^{-5} on both HM3D and Gibson environments. We compare three model checkpoints at 30k, 40k, and 50k and pick the one with the highest fine-tuning performance. Then we fine-tune DUET for 150k iterations, with batch size 32 and learning rate 2×10^{-5} on a single NVIDIA A100 GPU.

B. Additional Experiments (§4)

Here we provide additional experiments to investigate the effect of visual encoder, model initialization, and depth features. We also experiment with different model architectures (*i.e.*, HAMT [16]) on R2R dataset, and show object grounding results for the REVERIE task.

B.1. Effect of Visual Encoders (§4.2)

We study the effect of visual encoders in Table 12. Here we adopt CLIP’s ViT backbone with different model sizes and input patches (*i.e.*, Base/16, Large/14, and Huge/14). We can see that the vision encoder has a major influence on SPL, suggesting the agent can make fewer wrong steps and is capable of efficient navigation.

Visual Encoders	R2R Val-Seen				R2R Val-Unseen			
	TL	NE↓	SR↑	SPL↑	TL	NE↓	SR↑	SPL↑
CLIP-ViT-B/16	12.41	2.02	80.51	74.88	13.16	2.53	78.08	68.31
CLIP-ViT-L/14	12.62	2.16	80.04	74.06	13.13	2.50	78.08	68.97
CLIP-ViT-H/14	12.53	2.15	81.19	76.83	12.61	2.49	78.20	69.71

Table 12: Effect of visual encoders.

B.2. Effect of Initialization (§4.2)

Table 13 presents the performance of initializing the navigation agent with different pre-trained models in pre-training. We discovered that utilizing BERT to initialize the language encoder does not enhance downstream performance, and even harms the performance on the validation unseen set. We attribute this to the vast domain gap between uni-modal BERT’s language representations and CLIP’s visual representation. Results could be improved by initializing the model with LXMERT’s language encoder [73], and even more by utilizing both the language encoder and cross-modal encoder from LXMERT, indicating that incorporating pre-trained vision-and-language models could benefit agent performance.

Language Encoder Initialization	R2R Val-Seen				R2R Val-Unseen			
	TL	NE↓	SR↑	SPL↑	TL	NE↓	SR↑	SPL↑
Random	12.87	2.29	78.75	72.61	12.69	2.72	75.65	67.00
BERT	12.43	2.29	79.04	73.72	12.95	2.76	75.01	66.57
LXMERT (lang.)	11.73	2.07	80.22	75.65	13.17	2.67	75.86	67.36
LXMERT (lang.+cross.)	12.63	2.27	79.24	73.34	12.83	2.62	76.59	67.74

Table 13: Effect of different initialization, where *LXMERT (lang.)* means only initialize the language encoder with LXMERT, and *LXMERT (lang.+cross.)* means initialize both the language encoder and cross modal encoder with LXMERT.

B.3. Effect of Depth Modality (§4.2)

We also explored leveraging depth information to improve visual representations as described in Table 14. In line with previous methods such as [44, 42, 30, 5], we directly concatenate the depth features from DDPPPO [82] (a ResNet backbone pre-trained on PointGoal navigation with depth inputs) and the RGB features (from CLIP ViT-B/16) to create the visual representations. Our findings indicate that when not using HM3D as the augmented environment, the agent’s SR is significantly better if learning from the additional depth input. However, this conclusion changes when HM3D environments are involved: the agent’s SR with RGBD was slightly lower than with RGB-only. We suspect that as the data is scaled up with more visual observations and language instructions, the agent may not require additional depth information to assist decision-making.

HM3D Aug	Sensor	R2R Val-Seen				R2R Val-Unseen			
		TL	NE↓	SR↑	SPL↑	TL	NE↓	SR↑	SPL↑
×	RGB	13.28	2.51	76.89	69.71	13.53	3.06	72.92	62.82
	RGBD	14.16	2.54	77.18	69.76	15.14	3.02	74.12	62.54
✓	RGB	12.63	2.27	79.24	73.34	12.83	2.62	76.59	67.74
	RGBD	11.24	2.12	79.73	75.45	12.93	2.63	76.46	68.52

Table 14: Effect of adding depth modality.

B.4. ScaleVLN with Different VLN Models (§4.2)

To evaluate the generalization ability of our ScaleVLN dataset, we also apply the augmented data to train different VLN agents, including Seq2Seq [10], EnvDrop [74], and HAMT [16]. The HAMT model is pre-trained and fine-tuned with the same data and configurations as we pre-trained the DUET model, while we follow similar configurations of Seq2Seq and Envdrop to the original papers. All three agents are trained with the CLIP ViT-B-16 feature. The results are shown in Table 15. Compared to using only PREVALENT [28] for augmentation, All three models significantly benefit from incorporating the ScaleVLN dataset, with 12.2%, 3.8%, 5.5% absolute increase in SR for Seq2Seq, EnvDrop, and HAMT, respectively. This shows that ScaleVLN strengthens models’ generalization ability. Note that Seq2Seq and Envdrop perform better on Val-Seen when using PREVALENT, mainly caused by overfitting the training environments.

Model	Pre-Train Data	Fine-Tune Data	R2R Val-Seen			R2R Val-Unseen		
			NE↓	SR↑	SPL↑	NE↓	SR↑	SPL↑
Seq2Seq [10]	-	R2R, PREV	3.89	58.18	38.49	6.32	37.34	23.21
	-	R2R, ScaleVLN	4.78	49.85	36.32	5.20	47.51	34.81
Envdrop [74]	-	R2R, PREV	3.65	66.12	61.72	4.41	59.22	52.35
	-	R2R, ScaleVLN	3.70	65.23	59.06	3.99	63.01	54.93
HAMT [16]	R2R, PREV, ScaleVLN	R2R, PREV	2.58	74.93	71.52	3.69	64.90	60.11
		R2R	2.15	79.53	76.64	3.43	67.56	62.32
		R2R, ScaleVLN	2.43	76.40	73.30	3.07	70.46	65.12

Table 15: Influence of ScaleVLN on different VLN models.

B.5. ScaleVLN for Long-Horizon VLN (§4.2)

We evaluate the impact of our dataset on a long-horizon VLN dataset, R4R [35]. R4R extends the R2R dataset by concatenating two adjacent trajectories in R2R, resulting in longer navigation trajectories not biased by the shortest path prior. We directly fine-tune our pre-trained HAMT models from Table 15 on R4R. Compared to pre-training with only R2R and PREVALENT, adding our ScaleVLN dataset in the pre-training stage leads to a consistent gain, yielding +2.7% SR, +1.5% nDTW and +2.7% SDTW [33]. As suggested by the large improvement in nDTW between the ground-truth path and the executed path, our ScaleVLN data not only facilitate the model to reach the target but also follow the path described by the given instruction.

Pre-Train Data	Fine-Tune Data	R4R Val-Unseen				
		NE↓	SR↑	CLS↓	NDTW↑	SDTW↑
R2R, PREV	R4R	6.19	41.52	57.89	51.21	30.00
R2R, PREV, ScaleVLN	R4R	6.09	44.20	59.55	52.77	32.73

Table 16: Effect of ScaleVLN on learning R4R.

B.6. Leaderboard Results of R2R (§4.4)

We report the top seven submissions on the test-unseen leaderboard of R2R⁶ (Table 17). When ranking with success rate, we can see that (a) most methods have extremely low SPL (1%) due to using beam search to find the optimal paths. Even so, our single-run result (*EarlyToBed*) outperforms them by a large margin. When ranking with SPL (b), some methods pre-explored the test environments but their results are still much worse than ours. Apart from human followers, we are currently ranked first on the leaderboard.

Team	NE↓	SR↑	SPL↑	Team	NE↓	SR↑	SPL↑
human	1.61	86	76	human	1.61	86	76
EarlyToBed (ours)	2.27	80	70	EarlyToBed (ours)	2.27	80	70
LILY [◦]	2.54	78	1	TAHCX†	3.00	73	69
Airbert [◦]	2.50	78	1	Active Exploration†	3.30	70	68
Shortest-Path-Prior [◦]	3.55	74	1	sponge	3.26	71	67
UU_77	3.00	74	63	Auxiliary Reasoning†	3.96	68	65
TAHC [◦]	2.99	74	1	SE-Mixed	3.52	70	65

(a) Top 7 in SR.

(b) Top 7 in SPL.

Table 17: R2R leaderboard results (28.JUL.2023). [◦]: Beam search. †: Pre-exploration.

B.7. REVERIE Object Grounding Result (§4.4)

We report the success rate of remote object grounding (RGS) and its path length-weighted result (RGSPL). As shown in Table 18, ScaleVLN achieves state-of-the-art performance on object grounding task on the test leaderboard, comparable to the previous best method AutoVLN [17].

Models	REVERIE Val-Unseen				REVERIE Test-Unseen			
	SR↑	SPL↑	RGS↑	RGSPL↑	SR↑	SPL↑	RGS↑	RGSPL↑
SIA [50]	31.53	16.28	22.41	11.56	30.80	14.85	19.02	9.20
HAMT [16]	32.95	30.20	18.92	17.28	30.40	26.67	14.88	13.08
DUET [18]	46.98	33.73	32.15	23.03	52.51	36.06	31.88	22.06
AutoVLN [17]	55.89	40.85	36.58	26.76	55.17	38.88	32.23	22.68
DUET+ScaleVLN(ours)	56.97	41.84	35.76	26.05	56.13	39.52	32.53	22.78

Table 18: Object grounding performance on REVERIE.

C. Comparison of Navigability Graphs (§3.2)

We visualize the navigability graphs produced by AutoVLN [17] and our method for several HM3D environments in Figure 3. We can see that our graphs are denser, covering more regions, have viewpoints away from obstacles, and are fully traversable in open space.

D. Recover High Quality Images (§3.2)

As introduced in Main Paper §3.2, we apply the Co-Modulated GAN [87] to recover the corrupted images rendered from the HM3D and Gibson environments. Specifically, we first render a panorama of shape 512×1024 from

the 3D mesh at each viewpoint. Then, we crop four images of shape 512×512 centered at 0° , 90° , 180° and 270° of the panorama (with overlapping), and recover them separately. Note that, in VLN, the panoramic observation at a viewpoint is represented by 36 single-view images at 12 viewing angles and three elevations [10]. We directly extract their corresponding regions from the four recovered images to obtain these single-view images for pre-training an agent.

Table 19 visualizes the difference between the rendered images and our recovered images. First, we can see that our method can recover missing regions, including outdoor scenes such as sky and trees (Example 1 & 4) and indoor scenes such as floor and walls (Example 6). Besides, the recovered images usually have less blurry or distorted areas, and the object boundaries are much clearer and sharper. For instance, the ceiling light in Example 2, the chairs in Example 3, and the door frames in Example 5. Even for the highly corrupted images from Gibson (Examples 4–6), we can see that the method can still recover the scene to a reasonable quality.

⁶R2R test server: <https://eval.ai/web/challenges/challenge-page/97/leaderboard/270>.

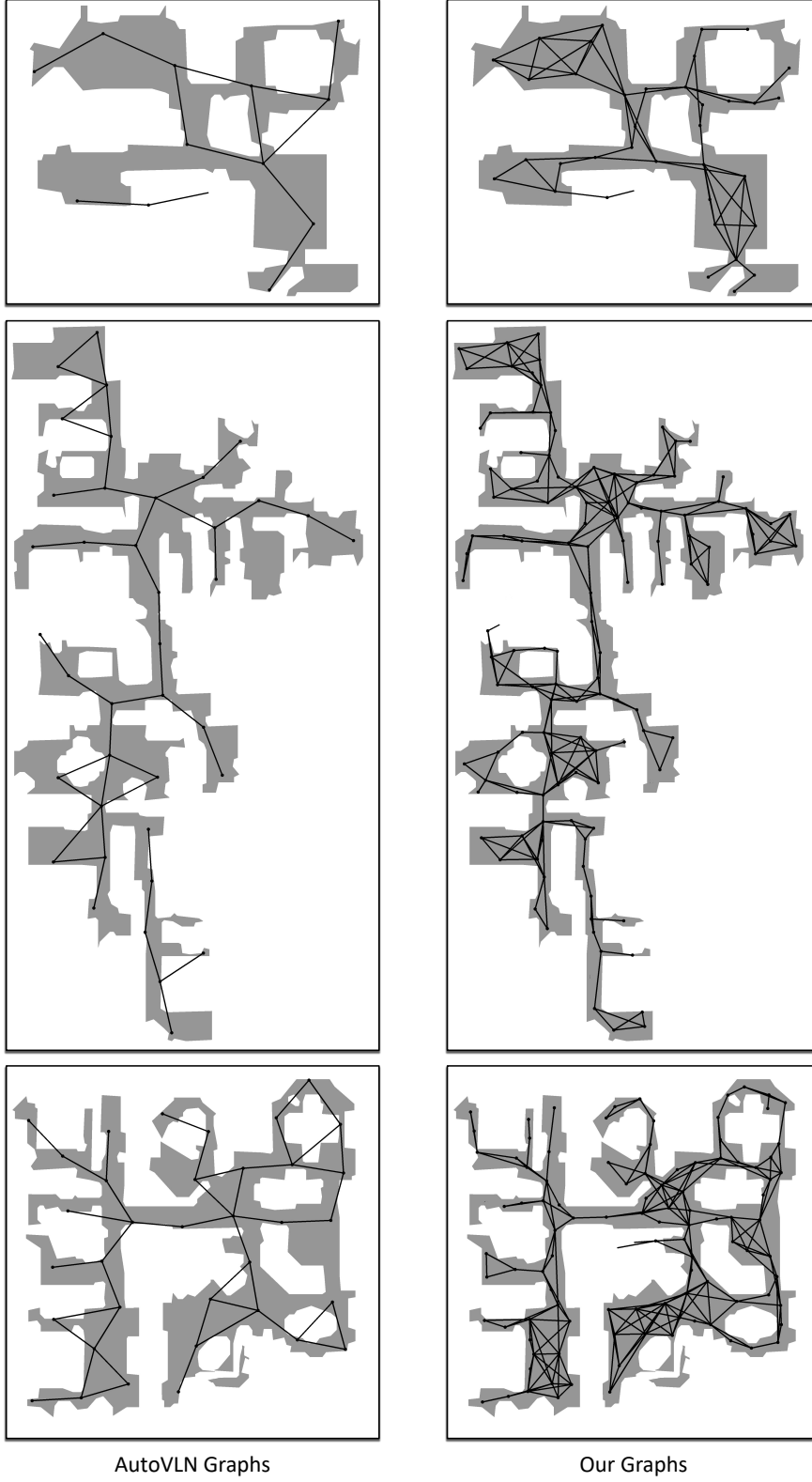


Figure 3: Comparison of navigability graphs between AutoVLN [17] and our ScaleVLN.

Examples	Environments	Rendered	Recovered
1	HM3D		
2	HM3D		
3	HM3D		
4	Gibson		
5	Gibson		
6	Gibson		

Table 19: Qualitative examples of our recovered images from HM3D and Gibson environments. The vertical line at the middle of panorama is caused by directly sticking two independently recovered images at 0° and 180° , which will not appear in the resulting augmented data, as explained in Appendix §D.