This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Improving Vision-and-Language Navigation by Generating Future-View Image Semantics

Jialu Li Mohit Bansal
UNC Chapel Hill
{jialuli, mbansal}@cs.unc.edu

Abstract

Vision-and-Language Navigation (VLN) is the task that requires an agent to navigate through the environment based on natural language instructions. At each step, the agent takes the next action by selecting from a set of navigable locations. In this paper, we aim to take one step further and explore whether the agent can benefit from generating the potential future view during navigation. Intuitively, humans will have an expectation of how the future environment will look like, based on the natural language instructions and surrounding views, which will aid correct navigation. Hence, to equip the agent with this ability to generate the semantics of future navigation views, we first propose three proxy tasks during the agent's in-domain pre-training: Masked Panorama Modeling (MPM), Masked Trajectory Modeling (MTM), and Action Prediction with Image Generation (APIG). These three objectives teach the model to predict missing views in a panorama (MPM), predict missing steps in the full trajectory (MTM), and generate the next view based on the full instruction and navigation history (APIG), respectively. We then fine-tune the agent on the VLN task with an auxiliary loss that minimizes the difference between the view semantics generated by the agent and the ground truth view semantics of the next step. Empirically, our VLN-SIG *achieves the new state-of-the-art on both Room-to-Room dataset and CVDN dataset. We further show that our agent learns to fill in missing patches in future views qualitatively, which brings more interpretability over agents' predicted actions. Lastly, we demonstrate that learning to predict future view semantics also enables the agent to have better performance on longer paths.*[1]

1. Introduction

In Vision-and-Language Navigation, the agent needs to navigate through the environment based on natural language instructions. Many datasets have been proposed to solve this challenging task [2, 5, 22, 33, 40]. Based on these datasets, previous works aim to strengthen the navigation model or agent from several aspects: understanding long and detailed instructions in different languages, inferring and locating target objects based on common knowledge, navigating in diverse and potentially unseen environments, and learning better alignment between the environment and the instructions. However, most previous work simplifies the navigation process as an action selection process, where at each time step, the agent picks the action from a set of pre-defined candidates. This simplification does not take advantage of human's ability to expect what scene will be more likely to happen next during navigation. For example, given the instruction "Walk through the bedroom into the hallway. Turn right and wait in the kitchen doorway.", before walking out of the bedroom, humans will expect the hallway to be a long passage with doors, and the kitchen probably contains objects like a kitchen island and sink. Humans have these expectations based on common sense knowledge and use them to select candidates during navigation. Thus, in this paper, we explore whether AI agents

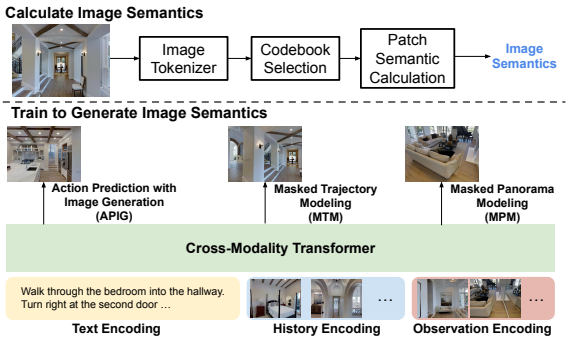

Figure 1. Overview of our proposed method VLN-SIG. We obtain the semantics of an image with a pre-trained image tokenizer, and use codebook selection (Sec. 3.4) and patch semantic calculation (Sec. 3.3) to adapt it to efficient in-domain VLN learning. We pre-train the agent on three proxy tasks (Sec. 3.5) and fine-tune the agent using Action Prediction with Image Generation (APIG) as the additional auxiliary task (Sec. 3.6).

[1]Code is available at https://github.com/jialuli-luka/VLN-SIG.git.

could also benefit from the ability to generate future scenes for action selection during navigation.

[21] first explores the useful idea of generating future scenes with high quality, based on history observations on an indoor Vision-and-Language Navigation dataset. Furthermore, they adapt their synthetic future scenes for VLN by replacing the original observations with the generated observations. However, their replacement method did not enhance the agents' performance on the VLN task. The potential of using semantic information from generated future observations is still underexplored. Thus, in this paper, we aim to equip the agent with both the ability to predict future scenes and also benefit from learning semantics in future generated observations.

We propose VLN-SIG: Vision-and-Language Navigation with Image Semantics Generation. As shown in Figure 1, to first calculate the overall image semantics, we tokenize the view images into visual tokens with a pre-trained discrete variational autoencoder (dVAE). While the pre-trained dVAE has a large vocabulary of 8192, we propose and compare static and dynamic codebook selection processes to optimize a subset of the codebook at one time during training. This helps the agent to learn the more important tokens and focus on optimizing more difficult tokens. Then, we propose and compare three ways based on mean value, block-wise weighted value, and sampling to represent the semantics of all patches for efficient image semantics learning. In the pre-training stage, we propose three tasks: Masked Panorama Modeling (MPM), Masked Trajectory Modeling (MTM) and Action Prediction with Image Generation (APIG). In Masked Panorama Modeling, the agent learns to predict the semantics of multiple missing views in a full panorama. In Masked Trajectory Modeling, the agent learns to predict the semantics of multiple steps in the full trajectory. MPM and MTM together give the agent the ability to understand the semantics in each visual token and learn to recognize the semantics contained in each view. In Action Prediction with Image Generation, the agent mimics the navigation process to predict the next step by generating the visual tokens contained in the next navigation view. This task enables the agent to imagine the next step view semantics before making actions. We then fine-tune the agent on the step-by-step Vision-and-Language Navigation task. We further enhance agents' ability to predict future views by optimizing an auxiliary task during navigation, which minimizes the difference between predicted observations and the target observation. Though this task does not help the agent make navigation decision directly, the future visual semantics injected by this task helps the agent understand the environment better.

We conduct experiments on Room-to-Room (R2R) datasets [2] and Cooperative Vision-and-Dialog Navigation (CVDN) dataset [40]. Empirical results show that our proposed VLN-SIG outperforms the strong SotA agents [8,10] by a relative gain of 4.5% in goal progress (meters) on CVDN test leaderboard, and 3% absolute gain in success rate on Room-to-Room test leaderboard. We further demonstrate that our proposed codebook selection methods and patch semantic calculation methods are crucial for learning to generate image semantics with ablation studies. Besides, we show that our agent achieves better performance for longer paths. Lastly, we show that our agent learns to fill in missing patches in the future views, which brings more interpretability over agents' predictions.

2. Related Work

Vision-and-Language Navigation. Vision-and-Language Navigation requires an agent to navigate based on natural language instructions. The navigation environment can be indoor [2,22,33,40], outdoor [5,31] or synthetic [29,32,37]. Previous work improves agents' navigation performance from several aspects. [17, 23, 24, 26, 36, 42, 44, 44] aim to learn better alignment between language, environment and actions, while [25, 39, 43] improve agents' generalization to unseen environments. Besides, graph information is incorporated as additional signals for pre-exploration or back-tracking to aid navigation [6,7,10,11,14,45]. Furthermore, pre-training techniques have been widely applied in VLN tasks, where [8,9,15,16,18,30,34] adapt pre-trained cross-modal representations or perform in-domain VLN pretraining to learn a better uni-modal and cross-modal representations. Different from previous agents which make action solely by picking from a set of candidates, we aim to equip the agent with the ability to generate semantics in future views to aid decision making. [21] proposes a separate image generation model to generate future scenes based on history observation in VLN. However, they use the generated future scenes as a substitution for the original environments, and their agent does not benefit from learning to navigate given generated scenes. Differently, we propose three new pre-training tasks that guide the agent to learn and generate the visual semantics in the observations, and further enhance the agent's ability to imagine the next step view semantics by optimizing an auxiliary task during fine-tuning.

Vision-and-Language Pre-training. Pre-training has been a popular technique to learn better uni-modal or multi-modal representations from large datasets. Many tasks have been proposed. For example, Masked Language Modeling (MLM) [12,27] is used for language-only pre-training, and Masked Region Modeling (MRM), and Image Text Matching (ITM) [28, 38] have been proposed for Vision-and-Language pre-training. Recently, [3] proposes Masked Image Modeling (MIM) for vision-only pre-training, where they mask several patches in an image and predict their corresponding visual tokens. MIM has been extended successfully to Vision-and-Language pre-training [4] and Video-

and-Language pre-training [13]. Inspired by them, we propose three VLN specific tasks based on MIM. To effectively and efficiently adapt MIM to VLN pre-training, we change the task objective to mask full images instead of patches in the image, and propose two ways for visual token codebook selection and three methods to represent overall image semantics.

3. Method

3.1. Problem Setup

Vision-and-Language Navigation (VLN) requires an agent to navigate through the environment based on natural language instructions. Formally, the agent navigates from start location B towards target location C based on language instructions I. At each navigation step, the agent has access to the panorama view P of the current location, which is discretized into 36 view images, and the agent will pick the next step from a set of K navigable locations $\{C_i\}_{i=1}^{K}$. The agent should predict a "STOP" action when it reaches the target location C.

3.2. Pre-training Overview

In the pre-training stage, the agent is pre-trained on six proxy tasks as in [8] and the three proxy tasks we propose in this paper. Specifically, the six tasks we adopt are Masked Language Modeling (MLM), Masked Region Modeling (MRM), Instruction Trajectory Matching (ITM), Single-step Action Prediction/Regression (SAP/SAR), and Spatial Relationship Prediction (SPREL). Details of these tasks can be found in Appendix.

Besides these, we further propose three tasks to help the agent learn to generate image semantics based on multimodal information to aid decision making during navigation (Figure 2). Formally, given an image $v_i \in \mathbb{R}^{H\times W\times C}$, we tokenize the image into N visual tokens $\{m_{ij}\}_{j=1}^{N}$ with an image tokenizer, where each visual token m_{ij} belongs to the vocabulary $T = \{1, ..., |T|\}$, and $N = HW/P^2$, P is the patch size. Following [3], we adopt the pre-trained discrete variational autoencoder (dVAE) from [35] to compute the probability $p_{ij} \in \mathbb{R}^{|T|}$ ($\sum_{k=1}^{|T|} p_{ijk} = 1$) for each visual token:

$$p_{ij1}, ..., p_{ij|T|} = \text{dVAE_Enc}(v_i) \tag{1}$$

Improving Efficiency by Generating Overall Image Semantics. Each visual token contains specific semantic information in the image. In Masked Image Modeling used in previous vision transformer pre-training [3,4], the model learns to predict the visual token for each masked patch in an image to learn the semantics for each patch. Different from them, we directly mask the full image and predict the overall semantics of the image. Our approach is *better in both memory cost and speed* for VLN tasks. First, by masking the full image, we are able to directly use the pre-extracted features and eliminate the need to tune the visual backbone that encodes images with masked patches. Our approach is more than two times faster than [8], which tunes the visual backbone end-to-end to enhance the agents' performance. Second, directly masking the full image and generating the overall semantics saves memory. It does not need to load the visual backbone in memory, and it takes the image representation as input, shrinking the input sequence length from $N * A$ (since previous MIM work [3] takes the patch representation as input) to A, where A is the sequence length of navigation history, and N is the number of patches in one image. To use our approach, it's important to have a good representation of the overall semantics of the image as the target. Thus, we explore and compare three ways to represent the probability of generating the overall semantics of the image in Sec. 3.3.

Furthermore, since the discrete variational autoencoder is learned from a large amount of text-image pairs collected from the internet, not every learned visual token in the vocabulary is of equal importance for the views in the Vision-and-Language Navigation environment. Thus, we propose two ways to optimize over a subset of visual tokens $S \subseteq T$ in Sec. 3.4.

Finally, we describe our three newly proposed pre-training proxy tasks that help the agent learn to generate the semantics of an image Sec. 3.5.

3.3. Overall Image Semantics Generation

In Vision-and-Language Navigation pre-training, the vision-modality input contains navigation history, which is represented as a sequence of A views. If we predict the generation of all masked patches in one image, it results in an input sequence length of $A * N$, which makes it computationally infeasible for VLN pre-training. Thus, we propose and compare three ways to efficiently predict overall image semantics generation probabilities.

Mean Patch Probability. The easiest way to represent the overall generation probability is to take the mean of all patches' probabilities. Specifically, given an image v_i, we represent the generation probability $p_{ik}^{o_m}$ as:

$$p_{ik}^{o_m} = \frac{1}{N}\sum_{j=1}^{N} p_{ijk} \tag{2}$$

where o indicates **o**verall generation probability and m indicates **m**ean patch probability.

Sample Patch Probability. Using the mean patch probability has one main drawback: when we optimize the KL-divergence loss between the predicted probability $\hat{p}_{ik}^{o_m}$ and the ground truth probability $p_{ik}^{o_m}$, it does not guarantee that

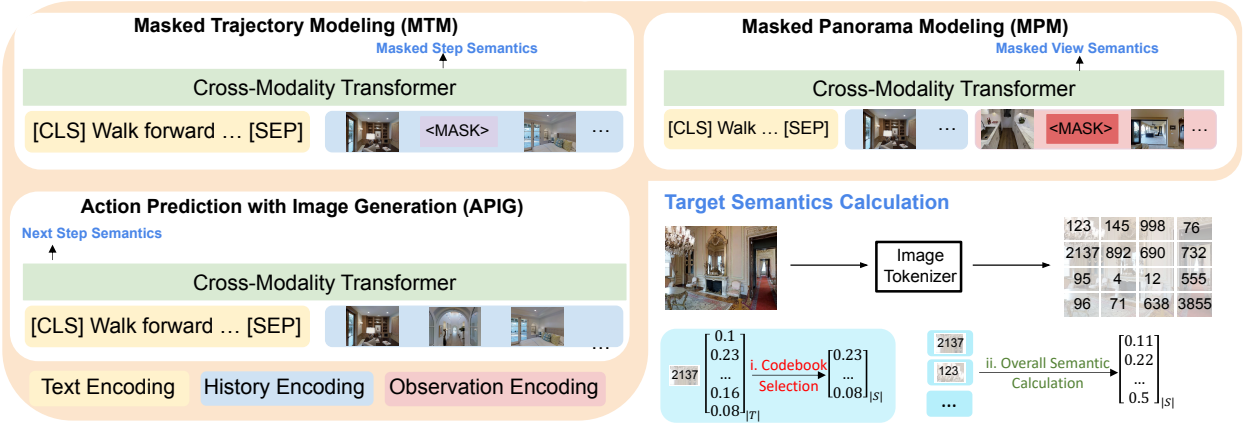

Figure 2. Illustration of the target semantics calculation process and our proposed three proxy tasks.

the agent learns the semantics of every patch (p_{ijk}) correctly. Formally, after optimization, our goal is that for every patch j in image v_i, $p_{ijk} - \hat{p}_{ijk} < \epsilon$, where ϵ is as small as possible. Directly optimizing the mean patch probability only minimizes the difference between $p_{ik}^{o_m}$ ($\frac{1}{N}\sum_{j=1}^{N} p_{ijk}$) and $\hat{p}_{ik}^{o_m}$ ($\frac{1}{N}\sum_{j=1}^{N} \hat{p}_{ijk}$), but for each patch j, p_{ijk} and $\hat{p}_{ijk}$ can still be different. Besides, as the model only learns to predict the mean patch probability, it cannot actually predict the token for each patch and generate the real images. To mitigate this problem, we propose sample patch probability. We sample one patch from N patches in an image, and use the semantics in the sampled patch to represent the image:

$$p_{ik}^{o_s} = p_{ijk} \tag{3}$$

where j is the sampled patch. We predict $p_{ik}^{o_s}$ based on encoded image inputs and sampled patch j's position. We encode the position of the patch with a learned position embedding, and add it to the encoded inputs.

Weighted Patch Probability. Sample patch probability aims to minimize the differences between p_{ijk} and $\hat{p}_{ijk}$ for every patch j in the image. However, minimizing the difference between p_{ij_1k} and $\hat{p}_{ij_1k}$ for patch j_1 might push p_{ij_2k} and $\hat{p}_{ij_2k}$ away for patch j_2, which makes the convergence harder. Thus, we further propose weighted patch probability. Formally, we represent image semantics generation probability $p_{ik}^{o_w}$ as:

$$p_{ik}^{o_w} = \sum_{j=1}^{N} w_j p_{ijk} \tag{4}$$

where w_j is a randomly sampled weight for patch j, and $\sum_{j=1}^{N} w_j = 1$. Since we randomly sample $\{w_j\}_{j=1}^{N}$ during training, after optimization, $\forall \{w_j\}_{j=1}^{N}, p_{ik}^{o_w} - \hat{p}_{ik}^{o_w} < \epsilon$. This constraint guarantees that for every patch j in the image v_i, $p_{ijk} - \hat{p}_{ijk} < \epsilon$ after optimization. Detailed proof can be found in Appendix. Furthermore, we propose block-wise weighted patch probability, where at each optimization step, we randomly sample a block which contains B patches, and the weights for the patches in the block are higher than other patches. The block-wise weights help the model learn to predict the richer semantics in the block compared with single token. We predict the weighted patch probability based on encoded image inputs and sampled weights, where we encode the sampled weights $\{w_j\}_{j=1}^{N}$ with fully-connected layers.

3.4. Visual Token Codebook Selection

Since the discrete variational autoencoder (dVAE) [35] is trained on image-text pairs from the web, not all the 8192 classes are of equal importance for the environment in VLN. Besides, it's hard to adapt 8192 classes to the VLN domain at the same time. Thus, we propose two ways to learn a subset of the vocabulary at each iteration during optimization.

Static Codebook Selection. We first propose a simple static codebook selection method, where we select a fixed set of $|S|$ visual tokens from the vocabulary T. Specifically, we calculate the visual token frequency in the training environment, and pick the $|S|$ visual tokens that have the highest frequency. During training, we only predict the generation probability of these static $|S|$ visual tokens. This static codebook selection filters out tokens that are rarely appeared in the dataset. Thus the agent could focus more on the frequent tokens.

Dynamic Codebook Selection. We propose a novel dynamic codebook selection method, where we dynamically select a set of S visual tokens from the vocabulary T. Specifically, the S tokens are initialized to be the tokens that have the highest frequency in the environments. During training, we update the frequency score s_f based on visual tokens that appear in the current training batch. Besides,

we add a difficulty score s_d based on the difference of the agents' predicted probability $\hat{p}^o_{ik}$ and the ground truth probability p^o_{ik} ($p^o_{ik} \in \{p^{o_m}_{ik}, p^{o_s}_{ik}, p^{o_w}_{ik}\}$). We pick the S tokens that have the highest score s_t at training step t:

$$s_d = |\hat{p}^o_{ik} - p^o_{ik}| \tag{5}$$

$$s_t = \lambda s_{t-1} + (1-\lambda)(s_f + \gamma s_d) \tag{6}$$

The score s_t takes both token frequency and learning difficulty into consideration. The agent learns to focus on more frequent visual tokens and also puts more effort into learning the more difficult tokens.

3.5. Pre-training Tasks

We propose three proxy tasks for the agent to learn to generate image semantics calculated with the codebook selection (Sec. 3.4) and patch semantic calculation (Sec. 3.3). We directly adopt the history-aware transformer model HAMT [8] as our base agent, and pre-train it from scratch with the 6 tasks proposed by [8] and our proposed 3 tasks.

Masked Trajectory Modeling (MTM). In Masked Trajectory Modeling (MTM) (Top Left in Fig. 2), we randomly mask out $r\%$ of the steps in the full navigation trajectory, and aim to recover those views based on language instructions and other views in the trajectory. Specifically, given language instruction $\{w_1, ..., w_L\}$ and navigation trajectory $\{v_1, ..., v_M\}$, we randomly replace some steps v_i with a $\langle MASK \rangle$ token. We predict $\hat{p}^o_{ik_{MTM}}$ ($\hat{p}^o_{ik_{MTM}} \in \{\hat{p}^{o_m}_{ik_{MTM}}, \hat{p}^{o_s}_{ik_{MTM}}, \hat{p}^{o_w}_{ik_{MTM}}\}$) with a multi-layer fully-connected network: $\hat{p}^o_{ik_{MTM}} = f_{MTM}(h_i + E_p)$, where h_i is the model output of the masked step v_i. E_p is zeros when we use mean patch probability, E_p is the position encoding of the sampled patch when using sample patch probability, and E_p is encoded sampled weights when using weighted patch probability. We optimize the KL-divergence loss between the two distributions:

$$L_{MTM} = -\sum_{k \in S} p^o_{ik} log \hat{p}^o_{ik_{MTM}} \tag{7}$$

Masked Panorama Modeling (MPM). At each time step, the agent observes a panorama of the current location. The panorama is discretized into 36 views. In Masked Panorama Modeling (Top Right in Fig. 2), we randomly mask out $u\%$ of the views in the panorama, and aim to predict the masked views based on language instructions, navigation history and surrounding views in the panorama. Specifically, given language instruction $\{w_1, ..., w_L\}$, navigation history $\{v_1, ..., v_M\}$, and current panorama view $\{v_{M+1_i}\}_{i=1}^{36}$, we randomly replace some views v_{M+1_i} with a <MASK> token. We predict $\hat{p}^o_{ik_{MPM}}$ with a multi-layer fully-connected network: $\hat{p}^o_{ik_{MPM}} = f_{MPM}(o_i + E_p)$, where o_i is the output of the masked discretized view. We optimize the KL-divergence loss between two distributions:

$$L_{MPM} = -\sum_{k \in S} p^o_{ik} log \hat{p}^o_{ik_{MPM}} \tag{8}$$

Action Prediction with Image Generation (APIG). Previous work picks the next action of the agent by directly selecting from a set of candidates. In this task, the agent mimics the action prediction process by generating the future scenes and picking the candidates that are closest to the generated images. In Action Prediction with Image Generation, we aim to generate semantics of the views at the next step, based on language instruction $\{w_1, ..., w_L\}$ and navigation history $\{v_1, ..., v_M\}$. We predict $\hat{p}^o_{ik_{APIG}}$ with a multi-layer fully connected network: $\hat{p}^o_{ik_{APIG}} = f_{APIG}(w_0 + E_p)$, where w_0 is the output of the <CLS> token in the instructions. The <CLS> token contains state information of the agent after cross-modality attention layers between the instructions and the navigation history. We optimize the KL-divergence loss between two distributions:

$$L_{APIG} = -\sum_{k \in S} p^o_{ik} log \hat{p}^o_{ik_{APIG}} \tag{9}$$

3.6. Fine-tuning on Navigation Task

We fine-tune the agent on the navigation task with a mixture of imitation learning (IL) and reinforcement learning (RL) as in [8]. To fully utilize agents' ability to generate semantics in future scenes, we further train the agent with APIG as an auxiliary task. We initialize the model weights for this task with f_{APIG} learned in pre-training. Since it's computationally inefficient to compute the ground truth generation probability for an image during VLN fine-tuning, we pre-extract the mean patch probability $p^{o_m}_{ik}$ of an image v_i and the target visual token ($m_{ij} = argmax_k(p_{ijk})$) for every patch j in an image v_i. For $p^{o_m}_{ik}$, we optimize the KL-divergence loss between two distributions: $L_{AT} = -\sum_{k \in S} p^{o_m}_{ik} log \hat{p}^{o_m}_{ik}$. For $p^{o_s}_{ik}$, since we don't pre-extract the visual token probability p_{ijk}, we optimize the cross-entropy loss between predicted generation probability $\hat{p}^{o_s}_{ik}$ and the sampled visual token class m_{ij} instead: $L_{AT} = -m_{ij} log \hat{p}^{o_s}_{im_{ij}}$. For p^w_{ik}, we fix the weights w_j to be $1/N$ to generate mean patch probability instead, and optimize the KL-divergence loss as: $L_{AT} = -\sum_{k \in S} p^{o_w}_{ik} log \hat{p}^{o_w}_{ik}$.

4. Experimental Setup

4.1. Dataset

We evaluate our agent on the Room-to-Room (R2R) dataset [2] and Cooperative Vision-and-Dialog Navigation (CVDN) dataset [40]. Both datasets are split into training,

Model	R2R				CVDN	
	Val Unseen		Test		Val Unseen	Test
	SR↑	SPL↑	SR↑	SPL↑	GP↑	GP↑
PREVALENT [16]	58	53	54	51	3.15	2.44
◯BERT [18]	63	57	63	57	-	-
HOP [34]	64	57	64	59	4.41	3.31
HAMT [8]	66	61	65	60	5.13	5.58
Ours	**68**	**62**	**65**	**60**	**5.52**	**5.83**
SSM [41]♠	62	45	61	46	-	-
DUET [10]♠	72	60	69	59	-	-
Ours♠	**72**	**62**	**72**	**60**	-	-

Table 1. Comparison with state-of-the-art agents on Room-to-Room (R2R) and Cooperative Vision-and-Dialog Navigation (CVDN) validation unseen set and test leaderboard. ♠ denotes agent that utilizes graph information during navigation.

No.	Pre-Training Tasks			Validation Unseen					
	MTM	MPM	APIG	NE↓	SR↑	SPL↑	nDTW↑	sDTW↑	CLS↑
1	✗	✗	✗	-	64.4	58.8	-	-	-
2	✓	✗	✗	3.52	66.7	61.3	68.7	57.3	67.3
3	✗	✓	✗	3.68	65.5	59.4	68.2	56.5	66.8
4	✗	✗	✓	3.58	66.3	60.9	68.6	57.2	67.0
5	✓	✓	✓	**3.37**	**68.1**	**62.3**	**69.6**	**58.7**	**67.7**

Table 2. Ablation results for our proposed three proxy tasks on Room-to-Room validation unseen set.

seen validation, unseen validation and test set. The environments in unseen validation and test set are not appeared in the training set. We focus on agents' performance on the unseen validation set and the test set, since they measure agents' ability to generalize to unseen environments.

4.2. Evaluation Metrics

We evaluate our model on the following metrics: (1) Success Rate (SR), whether the agent stops within 3 meters to the target. (2) Success Rate Weighted by Path Length (SPL) [1], which penalizes long paths that randomly explore the environment. (3) normalized Dynamic Time Warping (nDTW) [19], measures how well the agent follows the ground truth path. (4) success rate weighted by normalized Dynamic Time Warping (sDTW), which considers nDTW score for success cases. (5) Coverage weighted by Length Score (CLS) [20], measures how well the agent follows the reference path. (6) Navigation Error (NE), the distance between the stop location and the target. (7) Trajectory Length (TL), the total navigation length of the agent. We consider success rate as the main evaluation metric.

4.3. Implementation Details

We adopt the model architecture from [8]. For the image tokenizer, the input image size is 224, and the patch size is 16. We set 1.0 for γ and 0.5 for λ in dynamic codebook selection, and $|S|$ to be 1000 for both codebook selection methods. In pre-training, the ratio to select tasks is set to be 3 for MTM and 1 for others. The mask ratio r is 0.5 for MTM and u is 0.3 for MPM. In fine-tuning, L_{AT} is added to the IL loss with ratio 1. The ratio to combine IL and RL is 0.15 when adding L_{AT} and 0.2 otherwise. We use weighted patch probability with block-wise sampling for Room-to-Room dataset and base weighted patch probability for CVDN dataset. Other hyperparameters are the same as in [8] for fair comparison.

5. Results and Analysis

5.1. Test Set Results

We show our method's performance on both the Room-to-Room (R2R) and the Cooperative Vision-and-Dialog Navigation (CVDN) dataset. As shown in Table 1, our agent outperforms previous SotA agent HAMT [8] which utilizes computational expensive end-to-end visual backbone tunning by 0.25 in GP on CVDN test leaderboard, and shows competitive performance on R2R test leaderboard. This demonstrates that our agent benefits from imagining the future view semantics during navigation. Furthermore, we demonstrate that our proposed methods generalize well to graph-based VLN agent DUET [10], where the agent encodes the current location with neighboring nodes representation and makes action conditioned on both coarse-scale topological map encoding and fine-scale current step encoding. Specifically, we pre-train the agent with Masked Pano Modeling (MPM) and Action Prediction with Image Generation (APIG). Masked Trajectory Modeling (MTM) is not used in pre-training as [10] doesn't utilize explicit trajectory history step encoding for final action prediction. Our method outperforms DUET [10] by 3% in success rate and 1% in SPL, achieving the new state-of-the-art on Room-to-Room test leaderboard.

5.2. Effectiveness of Pre-training Tasks

In this section, we demonstrate the effectiveness of our proposed three tasks for VLN in-domain pre-training. We

No.	PT Tasks	Codebook Selection		Validation Unseen					
		Static	Dynamic	NE↓	SR↑	SPL↑	nDTW↑	sDTW↑	CLS↑
1	Baseline			-	64.4	58.8	-	-	-
2	MTM	✗	✗	3.66	65.1	59.8	68.5	56.7	66.5
3	MTM	✓	✗	3.65	66.2	60.3	67.5	56.7	66.3
4	MTM	✗	✓	3.52	66.7	61.3	68.7	57.3	67.3
5	MPM	✗	✗	3.89	64.8	59.0	68.0	55.4	65.3
6	MPM	✓	✗	3.68	65.5	59.4	68.2	56.5	66.8
7	MPM	✗	✓	3.64	65.1	59.8	68.0	56.3	66.2
8	APIG	✗	✗	3.87	64.4	59.2	67.1	55.4	65.6
9	APIG	✓	✗	3.76	65.9	60.3	67.4	56.1	65.6
10	APIG	✗	✓	3.58	66.3	60.9	68.6	57.2	67.0

Table 3. Comparison of training without codebook selection, training with static codebook selection method and dynamic codebook selection method on Room-to-Room validation unseen set.

No.	PT Tasks	Semantic Calculation				Validation Unseen					
		Mean	Sample	Weighted	Weighted-Block	NE↓	SR↑	SPL↑	nDTW↑	sDTW↑	CLS↑
1	Baseline					-	64.4	58.8	-	-	-
2	ALL	✓	✗	✗	✗	3.66	65.5	60.3	68.8	56.4	67.4
3	ALL	✗	✓	✗	✗	3.67	65.9	60.2	68.0	56.6	66.5
4	ALL	✗	✗	✓	✗	3.43	66.9	61.8	**69.7**	58.2	**68.4**
5	ALL	✗	✗	✗	✓	**3.37**	**68.1**	**62.3**	69.6	**58.7**	67.7

Table 4. Comparison of different ways to calculate image semantics on Room-to-Room validation unseen set. "Weighted" indicates using weighted patch probability without block-wise sampled weights, and "Weighted-Block" samples block-wise weights when calculating weighted patch probability.

show the ablation performance of using one of the tasks in pre-training in Table 2. Specifically, the baseline model is HAMT pre-trained on 6 proxy tasks: MLM, MRM, ITM, SAP/SAR, and SPREL without end-to-end vision-backbone tunning (details in [8]). The performance shown in this table is the best model picked among the different combinations of codebook selection methods and patch probability calculation methods. As shown in Table 2, adding any of our proposed tasks improves the performance by a large margin in both success rate (SR) and success rate weighted by path length (SPL). Specifically, adding Masked Trajectory Modeling (MTM) brings the largest improvement, outperforms the baseline by 2.3% in SR and 2.5% in SPL. Furthermore, pre-training on all the tasks achieves the best performance, improving the baseline by 3.7% in SR and 3.5% in SPL, demonstrating our approach's effectiveness.

5.3. Effectiveness of Codebook Selection

We demonstrate that it's important to select a subset of visual tokens either statically or dynamically for the agent to effectively learn the large visual token vocabulary extracted with pre-trained dVAE. We show the ablation performance in Table 3. First, we observe that the agent cannot learn useful image semantics when training on MTM or APIG without codebook selection (Model 1 vs. 2,8), but using dynamic codebook selection could significantly improve the performance, outperforming the model without codebook selection by 1.6% in SR for MTM (Model 2 vs. 4) and 1.9% in SR for APIG (Model 8 vs. 10). For MPM, using static codebook selection slightly outperforms the model that directly learns from all visual tokens in the vocabulary. We attribute this to that reasoning the masked view in a panorama is easier than generating the semantics in a masked step or next view. The agent could easily infer semantics from the overlaps between continuous views, and thus codebook selection benefits less for MPM.

5.4. Comparison of Image Semantic Representation Methods

We compare the three ways of representing the overall image semantics in Table 4. First, we observe that the agent benefits from pre-training with our proposed three tasks, regardless of how we calculate the image semantics (Model 1 vs. 2,3,4,5). Then, we observe that using sample patch probability, weighted patch probability without block-wise sampling, and weighted patch probability with block-wise sampling outperforms the mean patch probability (Model 2 vs. 3,4,5) by more than 0.5%, 1.4%, and 2.6% in SR respectively. This demonstrates that our proposed sample patch probability and weighted patch probability (with and without block-wise sampling) help the agent learn from semantics in each patch better, while saving the memory by $O(N)$ compared with traditional Masked Image Modeling method [3] which generates the probability for all masked patches in an image. Lastly, we demonstrate that adding block-wise sampling for the weights during training (Model 4 vs. 5) further improves the performance by 1.2% in success rate, demonstrating the importance of learning semantics in a block instead of sparsely at each optimization step.

No.	PT Tasks	Codebook Selection		FT Tasks	Validation Unseen					
		Static	Dynamic	APIG	NE↓	SR↑	SPL↑	nDTW↑	sDTW↑	CLS↑
1	Baseline				-	64.4	58.8	-	-	-
2	APIG	✓	✗	✗	3.64	64.9	60.4	67.9	56.2	66.7
3	APIG	✓	✗	✓	3.76	65.9	60.3	67.4	56.1	65.6
4	APIG	✗	✓	✗	3.53	66.1	60.7	68.1	56.9	66.5
5	APIG	✗	✓	✓	3.58	66.3	60.9	68.6	57.2	67.0

Table 5. Ablation results for adding Action Prediction with Image Generation (APIG) as auxiliary task during fine-tuning on Room-to-Room validation unseen set.

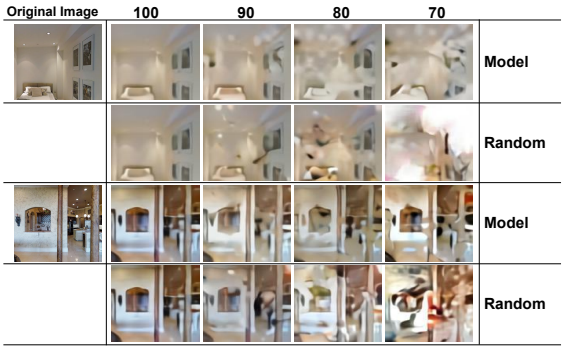

Figure 3. Qualitative analysis of images generated with our agent. Given 90%/ 80%/ 70% ground truth tokens, the rest tokens predicted by our model contain closer semantic information compared with randomly filled tokens.

5.5. Effectiveness of Auxiliary Task during Fine-tuning

We demonstrate the effectiveness of utilizing future semantics generation as auxiliary task during fine-tuning in this section. We use mean patch probability to represent the image semantics in these experiments. As shown in Table 5, adding APIG as auxiliary task during fine-tuning improves the performance by 1.0% in SR when using static codebook selection, and 0.2% in SR when using dynamic codebook selection. This shows that the agent can benefit from generating semantics in future views during navigation.

5.6. Ability to Generate Future Views

We demonstrate that our model could reasonably generate future semantics and reconstruct future images. We generate the token for each patch in the image with our APIG head learned with weighted patch probability. We set the weight vector to be an indicator vector, where the weight for the target patch is 1 and the others are 0. We use $x\%$ ($x \in 70, 80, 90$) of the patches with ground truth tokens encoded with pre-trained dVAE, the rest patches with the tokens generated by our APIG head, and then use the dVAE decoder to decode the image. As shown in Figure 3, our generated image could almost reconstruct the beds and pictures in the original image with small vague areas when given 70% of the ground truth tokens. In comparison, filling the 30% patches with random tokens will generate distorted images with large white parts, and the beds and pictures information cannot be identified. We also notice that our model still fails to generate the full image when all the tokens in the image are predicted by our APIG head, and needs at least 70% of the ground truth tokens to generate images of reasonable quality. More examples in appendix.

Path Length	3	4	5	6
HAMT [8]	66.7	65.8	67.5	62.3
Ours	66.7	**67.7**	**70.3**	**66.5**

Table 6. Success rate of our method and the baseline method for navigation trajectories with different lengths on Room-to-Room validation unseen set. Our method achieves larger improvement for longer paths.

5.7. Better Performance for Longer Trajectories

We show that our model is better at navigating longer paths on validation unseen set. Room-to-Room validation unseen set contains path lengths ranging from 3 to 6. As shown in Table 6, our model improves the baseline by 4.2% in success rate for paths with length 6, 2.8% for paths with length 5, and 1.9% for paths with length 4, demonstrating that learning to predict future view semantics can help the agent to learn to navigate longer paths better.

6. Conclusion

In this paper, we propose VLN-SIG, which explores whether navigation agents can benefit from imagining semantics in future views. We first propose novel codebook selection and image semantic calculation methods to efficiently learn the image semantics. We then design Masked Trajectory Modeling and Masked Panorama Modeling to help the agent to understand semantics in each visual token and learn to recognize the semantics contained in each view. We further use Action Prediction with Image Generation to enhance agents' ability to predict future views in both pre-training and fine-tuning. Our VLN-SIG achieves the new SotA on both CVDN dataset and R2R dataset.

Acknowledgements

We thank the reviewers for their helpful comments. This work was supported by ARO Award W911NF2110220 and DARPA MCS N66001-19-2-403. The views contained in this article are of the authors and not of the funding agency.

References

[1] 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. 6

[2] 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, 5

[3] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *ICLR*, 2022. 2, 3, 7

[4] Hangbo Bao, Wenhui Wang, Li Dong, and Furu Wei. Vl-beit: Generative vision-language pretraining. *arXiv preprint arXiv:2206.01127*, 2022. 2, 3

[5] Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12538–12547, 2019. 1, 2

[6] Kevin Chen, Junshen K Chen, Jo Chuang, Marynel Vázquez, and Silvio Savarese. Topological planning with transformers for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11276–11286, 2021. 2

[7] 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. 2

[8] 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, 2021. 2, 3, 5, 6, 7, 8

[9] Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. Learning from unlabeled 3d environments for vision-and-language navigation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIX*, pages 638–655. Springer, 2022. 2

[10] 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, 6

[11] 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

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *NAACL*, 2019. 2

[13] Yuying Ge, Yixiao Ge, Xihui Liu, Alex Jinpeng Wang, Jianping Wu, Ying Shan, Xiaohu Qie, and Ping Luo. Miles: Visual bert pre-training with injected language semantics for video-text retrieval. *arXiv preprint arXiv:2204.12408*, 2022. 3

[14] Georgios Georgakis, Karl Schmeckpeper, Karan Wanchoo, Soham Dan, Eleni Miltsakaki, Dan Roth, and Kostas Daniilidis. Cross-modal map learning for vision and language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15460–15470, 2022. 2

[15] 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. 2

[16] 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. 2, 6

[17] 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*, 33:7685–7696, 2020. 2

[18] Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. A recurrent vision-and-language bert for navigation. *CVPR*, 2021. 2, 6

[19] Gabriel Ilharco, Vihan Jain, Alexander Ku, Eugene Ie, and Jason Baldridge. General evaluation for instruction conditioned navigation using dynamic time warping. *Neural Information Processing Systems (NeurIPS)*, 2019. 6

[20] Vihan Jain, Gabriel Magalhaes, Alexander Ku, Ashish Vaswani, Eugene Ie, and Jason Baldridge. Stay on the path: Instruction fidelity in vision-and-language navigation. *ACL*, 2019. 6

[21] Jing Yu Koh, Honglak Lee, Yinfei Yang, Jason Baldridge, and Peter Anderson. Pathdreamer: A world model for indoor navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14738–14748, 2021. 2

[22] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-Across-Room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *Conference on Empirical Methods for Natural Language Processing (EMNLP)*, 2020. 1, 2

[23] Federico Landi, Lorenzo Baraldi, Marcella Cornia, Massimiliano Corsini, and Rita Cucchiara. Multimodal attention networks for low-level vision-and-language navigation. *Computer Vision and Image Understanding*, 210:103255, 2021. 2

[24] Jialu Li, Hao Tan, and Mohit Bansal. Improving cross-modal alignment in vision language navigation via syntactic information. *NAACL*, 2021. 2

[25] 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. 2

[26] Bingqian Lin, Yi Zhu, Zicong Chen, Xiwen Liang, Jianzhuang Liu, and Xiaodan Liang. Adapt: Vision-language navigation with modality-aligned action prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15396–15406, 2022. 2

[27] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 2

[28] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019. 2

[29] Matt MacMahon, Brian Stankiewicz, and Benjamin Kuipers. Walk the talk: Connecting language, knowledge, and action in route instructions. *Def*, 2(6):4, 2006. 2

[30] 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 *European Conference on Computer Vision*, pages 259–274. Springer, 2020. 2

[31] Harsh Mehta, Yoav Artzi, Jason Baldridge, Eugene Ie, and Piotr Mirowski. Retouchdown: Adding touchdown to streetlearn as a shareable resource for language grounding tasks in street view. *Proceedings of the Third International Workshop on Spatial Language Understanding*, 2020. 2

[32] Dipendra Misra, Andrew Bennett, Valts Blukis, Eyvind Niklasson, Max Shatkhin, and Yoav Artzi. Mapping instructions to actions in 3d environments with visual goal prediction. *EMNLP*, 2018. 2

[33] 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. 1, 2

[34] 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. 2, 6

[35] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 3, 4

[36] 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. *EMNLP*, 2021. 2

[37] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10740–10749, 2020. 2

[38] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *EMNLP*, 2019. 2

[39] Hao Tan, Licheng Yu, and Mohit Bansal. Learning to navigate unseen environments: Back translation with environmental dropout. *NAACL*, 2019. 2

[40] Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. Vision-and-dialog navigation. In *Conference on Robot Learning*, pages 394–406. PMLR, 2020. 1, 2, 5

[41] 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. 6

[42] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 6629–6638, 2019. 2

[43] 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 *European Conference on Computer Vision*, pages 413–430. Springer, 2020. 2

[44] Yue Zhang and Parisa Kordjamshidi. Explicit object relation alignment for vision and language navigation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 322–331, 2022. 2

[45] 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