Towards Navigation by Reasoning over Spatial Configurations

Yue Zhang
Michigan State University
zhan1624@msu.edu

Quan Guo
Michigan State University
guoquan@msu.edu

Parisa Kordjamshidi
Michigan State University
kordjams@msu.edu

Abstract

We deal with the navigation problem where the agent follows natural language instructions while observing the environment. Focusing on language understanding, we show the importance of spatial semantics in grounding navigation instructions into visual perceptions. We propose a neural agent that uses the elements of spatial configurations and investigate their influence on the navigation agent's reasoning ability. Moreover, we model the sequential execution order and align visual objects with spatial configurations in the instruction. Our neural agent improves strong baselines on the seen environments and shows competitive performance on the unseen environments. Additionally, the experimental results demonstrate that explicit modeling of spatial semantic elements in the instructions can improve the grounding and spatial reasoning of the model.

1 Introduction

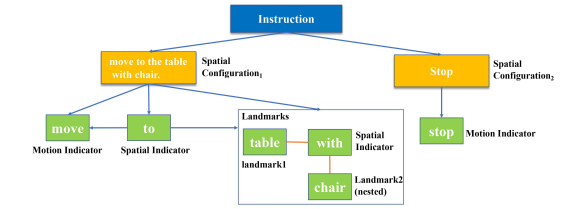
The ability to understand and follow natural language instructions is critical for intelligent agents to interact with humans and the physical world. One of the recently designed tasks in this direction is Vision-and-Language Navigation (VLN) (Anderson et al., 2018), which requires an agent to carry out a sequence of actions in a photo-realistic simulated environment in response to a sequence of natural language instructions. To accomplish this task, the agent should have three abilities: understanding linguistic semantics, perceiving the visual environment, and reasoning over both modalities (Zhu et al., 2020; Wang et al., 2019). While understanding vision and language are difficult problems by themselves, learning the connection between them without direct supervision makes this task even more challenging (Hong et al., 2020).

To address this challenge, some neural agents establish the connection using attention mechanism to relate the tokens from a given instruction to the images in a panoramic photo (Anderson et al., 2018; Fried et al., 2018; Ma et al., 2019; Yu et al., 2018). Surprisingly, although those models can improve the performance, Hu et al. (2019) found they ignore the visual information. There is no clear evidence that the agent can correspond the components of the visual environment to the instructions (Hong et al., 2020). Based on these results, recent research started to improve the agent's reasoning ability by explicitly considering the structure of language and image. From the language side, Hong et al. (2020) annotated fine-grained sub-instructions and their corresponding trajectories and used the co-grounded features of a part of instruction and the image to predict the next action. From the image side, Hu et al. (2019) induced a high-level object-based visual representation to ground the language into the visual context.

In the same direction, we propose a neural agent, namely *Spatial-Configuration-Based-Navigation (SpC-NAV)*, and consider the structure of both modalities, that is, spatial semantics of the instructions and the objects in the images. We use the notion of *Spatial Configuration* (Dan et al., 2020) to model the instructions and design a state attention to ensure the execution order of spatial configurations. Then, we utilize the spatial semantics elements, namely *motion indicator*, *spatial indicator* and *landmark* in spatial configuration to establish the connection with the visual environment. Specifically, we use the similarity score between the landmark representation in the spatial configurations and the object representation in the panoramic images to control the transitions between configurations. Also, we align object representations with the configuration representations enriched with motion indicator, spatial indicator and landmark representations to finally select the navigable image.

A spatial configuration is the smallest linguistic

Proceedings of Second International Combined Workshop on Spatial Language Understanding andGrounded Communication for Robotics, pages 42–52
August 5–6, 2020.



(a) Spatial Configuration Scheme

(b) Spatial Configuration Annotation

Figure 1: **Spatial Configuration example**. The instruction "Move to the table with chair, and stop." can be split into two spatial configurations: "move to the table with chair" and "stop". In configuration1, "move" is motion indicator; "to" is spatial indicator; "table" is landmark. "table with chair" is a nested spatial configuration of configuration1. The role of "table" is trajector; "with" is spatial indicator; and "chair" is landmark. In configuration2, "stop" is motion indicator.

unit that describes the location/trans-location of an object with respect to a reference or a path that can be perceived in the environment. It contains fine-grained spatial roles, such as motion indicator, landmark, spatial indicator, trajector. Essentially, each spatial configuration forms a sub-instruction in our setting. Figure 1 shows an example of splitting an instruction into its corresponding spatial configurations and the extracted spatial roles. Previous research argues representing the semantic structure of the language could improve the reasoning capabilities of deep learning models (Dan et al., 2020; Zheng and Kordjamshidi, 2020). There are relevant work modeling the meaning of spatial semantics in probabilistic models (Kollar et al., 2010; Tellex et al., 2011) and neural models (Regier, 1996; Ghanimifard and Dobnik, 2019). However, its impact on deep learning models for navigation remains an open research problem.

The contribution of this paper is as follows:

1. We consider the spatial semantic structure of the instructions explicitly in terms of spatial configurations and their spatial semantic elements, i.e., spatial/motion indicators, and landmarks to enrich the configuration representations.
2. We introduce a state attention to guarantee that configurations are executed sequentially. Also, we utilize the grounding between the extracted spatial elements and the object representation to help control the transitions between configurations.
3. Our experiment results show that considering the explicit representation of semantic elements of the spatial configurations improves the strong baselines significantly in the seen environments and yields competitive results in the unseen environments.

2 Related Work

Older studies on navigation before the deep learning era are mostly symbolic grounding methods, which are based on parsing the semantics of the instruction and learning probabilistic models. MacMahon et al. (2006) used the parser to associate the linguistic elements in free-form instruction to their corresponding action, location and object in the environment. Tellex et al. (2011) represented the spatial language as a hierarchy of Spatial Description Clauses (SDC) and proposed a discriminative probabilistic graphical model to find the most probable path with the extracted SDC and the detected visual landmark. Mei et al. (2016) provided a good overview of the past classical work on navigation. However, one of the biggest limitations of those methods is that they required prior linguistic structure and manual annotations.

In recent years, given the new capabilities created by deep learning architectures, the navigation task is extended to the photo-realistic simulated environments (Anderson et al., 2018; Thomason et al., 2019; Chen et al., 2019). Based on this, a Sequence-to-Sequence (Seq2seq) baseline model was proposed by Anderson et al. (2018) to encode the instructions and decode the embeddings to identify the corresponding output action sequence with the observed images. Fried et al. (2018) proposed to train a speaker model to augment the instructions for the follower model. Ma et al. (2019) introduced a visual and textual co-attention mechanism and a progress monitor loss to track the execution progress. Although those agents achieved better performance, the semantic structures on both language and vision sides were ignored.

We aim to exploit both symbolic grounding and neural models in the spatial domain. Regier (1996) designed the neurons to learn the meaning of spatial prepositions. Ghanimifard and Dobnik (2019) explored the effects of spatial knowledge in a generative neural language model for the image description. We mainly work on incorporating the spatial semantics in navigation neural agent. Hong et al. (2020) recently provided a method to segment the long instruction into sub-instructions. They used a shifting attention module to infer whether the current sub-instruction has been completed. Sub-instructions differ from us as they manually aligned

the instructions and viewpoints to learn the alignments, while we modeled spatial semantics to guide the alignment automatically. Moreover, their proposed shifting attention module is hard attention, and a threshold is set to decide whether the agent should execute the next sub-instruction. However, we utilize the grounding between the landmarks and the objects to control the transitions between sub-instructions.

3 Navigation Model

3.1 Problem Formulation

In this task, the agent follows an instruction to navigate from a start viewpoint to a goal viewpoint in a photo-realistic environment. Formally, the agent is given a natural language instruction S, which is a sequence of tokens, and $\{s_1, s_2, \cdots\}$ is its corresponding token embeddings. The agent observes a 360-degree panoramic view of its surrounding scene at the current viewpoint. Here, we follow Ma et al. (2019) to map the n navigable viewpoints to discrete images from the current panoramic view[1]. We obtain n images corresponding to each navigable viewpoint $I = \{I_1, I_2, \cdots, I_n\}$. The task is to select the next viewpoint among the navigable viewpoints or the current viewpoint (indicating the stop), and finally, to generate the trajectory that takes the agent close to an intended goal location.

3.2 Sequence-to-Sequence

We model the agent with a LSTM-based sequence-to-sequence architecture (Sutskever et al., 2014) to control the flow of information, as illustrated in Fig 2. The encoder computes a contextual embedding $\bar{s}_j$ of each token embedding s_j in S by $\bar{s}_j = LSTM_{encode}(s_j)$. At each step t of navigation, the decoder receives the grounded instruction representation C_t^* and the aligned image representation I_t^* to update its context h_t by $h_t = LSTM_{decode}([C_t^*, I_t^*])$. Finally, we predict the probability distribution of the next navigable viewpoint p_t by h_t. We introduce the method to obtain C_t^* and I_t^* in Section 3.5 and Section 3.6, as well as the next viewpoint prediction in Section 3.7.

3.3 Spatial Configurations Representation

To obtain the configurations in a navigation instruction, we first split the instructions into sentences. Then we design a parser with rules applied on an off-the-shelf dependency parser[2] to extract all the verb phrases and noun phrases in each sentence. In general, each configuration contains at most one motion indicator. Since we aim to process instructions and look for motions, we split the sentences with the extracted verb phrases as motion indicators to obtain spatial configurations. We do not separate the nested configurations with no motion indicator and keep them attached to the dynamic configurations (i.e. the ones with motion-indicator). As shown in Figure 1, "table with chair" is the nested spatial configuration of "move to the table with chair". Here, we only consider the prepositions that are attached to verbs, and merge the spatial indicators and motion indicators such as "move to" and use them together as the motion indicator. After that, we insert a pseudo delimiter token after each configuration and identify their contained noun phrases as landmarks. Each navigation instruction S is split into m configurations. We re-organize the contextual embeddings of tokens $[\bar{s}_1, \bar{s}_2, \cdots]$ generated by the encoder into the array of spatial configurations representation $C = [C_1, C_2 \dots C_m]$, where m is the number of configurations in the instruction. In the i-th configuration representation $C_i = \left[c_1^i, c_2^i \cdots, c_{\mathrm{P}}^i\right]$, the j-th element c_j^i is the contextual embedding of the corresponding k-th tokens in the instruction: $c_j^i = \bar{s}_k$. The last token of each configuration is always the pseudo delimiter indexed by P, which contains the most comprehensive context information about the preceding words. Soft attention is widely used to merge a collection of representations V into one by weighted sum based on the relevance indicated by their associated keys representations K and a query Q, calculated by Eq. 1.

$$\text{SoftAttn}(Q; K; V) = \text{softmax}\left(\frac{Q^T W K}{\sqrt{d_k}}\right) V \tag{1}$$

where W is a trainable linear mapping, and d_k is the dimension of each representation in K. We apply a soft attention to each configuration representation with the pseudo delimiter representation c_{P}^i, which can be calculated by Eq. 2.

$$\bar{C}_i = \text{SoftAttn}_{\text{config}}(Q = c_{\mathrm{P}}^i; K = C_i; V = C_i) \tag{2}$$

After obtaining configuration representations, an agent needs to identify which configuration to fol-

[1] 12 headings and 3 elevations with 30 degree interval.

[2] https://spacy.io/

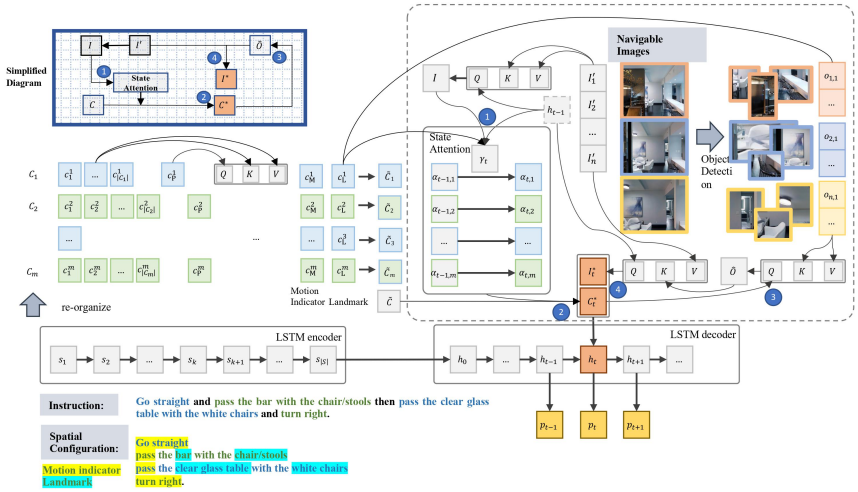


Figure 2: **Model Architecture.** The input to the encoder is the instruction text. The inputs to the decoder are the grounded language C_t^* calculated by state attention and the aligned visual representations I_t^* obtained from navigable images at each step t. The decoder predicts the distribution of next viewpoint p_t with the updated context h_t. The high-level view at the top-left shows the information flow in the model aligning with the circled numbers.

low at each step. To achieve this, we incorporate the intra-configuration and inter-configuration knowledge. Concretely, intra-configuration knowledge is the motion indicator that guides the agent movement and the landmarks that could be grounded into the objects in visual images; inter-configuration knowledge is that configurations should be processed one after another.

As mentioned above, we identify verbs and noun chunks in configurations as motion indicator and landmarks respectively. Each configuration can contain only one motion indicator and multiple landmarks. Formally, for the i-th configuration C_i, the motion indicator representation is denoted as c^i_{M} and the landmark representation is denoted as $c^i_L = \left[c^i_{\mathrm{L}_1}, c^i_{\mathrm{L}_2}, \cdots, c^i_{\mathrm{L}_p}\right]$, where p is the number of landmarks. If there is no landmark in the configuration, the value of c^i_{L} will be set as zeros. To enhance the motion indicator and landmark information, we concatenate their word embedding with the configuration representation. In case there are multiple noun chunks in configuration, to simplify, we select the noun closest to the root of the parsing tree as the main landmark, denoted as $\hat{p}$. Then the enriched configuration representation is denoted as $\tilde{C}_i = \left[\bar{C}_i; c^i_{\mathrm{M}}; c^i_{\mathrm{L}_{\hat{p}}}\right]$.

3.4 Visual Representation

To execute a series of configurations, the agent needs to keep track of the sequence of images observed along the navigation trajectory. We firstly transform the low-level image features from ResNet of n navigable images $I = \{I_1, I_2, \ldots, I_n\}$ to $I' = \left[I'_1, I'_2, \cdots, I'_n\right]$ by a fully-connected layer $I'_j = \mathrm{FC}_{\mathrm{img}}(I_j)$. Then, a soft attention is applied to I' with the previous context h_{t-1}, as shown in Eq. 3.

$$\bar{I} = \mathrm{SoftAttn}_{\mathrm{img}}(Q = h_{t-1}; K = I'; V = I') \tag{3}$$

Furthermore, we equip the agent with object-based representation. Specifically, we get top-K object representations from each image with an object detection model[3]. In this paper, we consider two kinds of object representation: object label representation and object visual representation. Specifically, the label representation uses the GloVe embedding (Pennington et al., 2014) of the type of the object, and visual representation uses the region-of-interest (ROI) pooling of the object detection model. We will compare the two representations and a hybrid representation of them in Appendix A.1. Formally, the object representations could be denoted as $O = [O_1, O_2 \ldots O_n]$, where for image I_j, there is $O_j = [o_{j,1}, o_{j,2}, \cdots, o_{j,K}]$. $o_{j,k}$ is the k-th object representation in j-th image.

3.5 Spatial Configuration Grounding

To guarantee the sequential execution, we design a state attention mechanism over the configurations.

[3]We employ Faster R-CNN pre-trained on Visual Genome, and use at most 36 objects that have an area greater than 10 pixels.

We consider the attention weight at each step as a state that measures navigation progress and is updated by a controller. Formally, the i-th configuration at step t is denoted as $\alpha_{t,i}$. At the first step, the attention weight is initialized to be focused on the first configuration $\alpha_0 = [1, 0, \cdots]$. At each of the following steps, the attention weight is updated by a controller γ_t with discrete convolution. γ_t is a two dimensional probability distribution indicating to what extent the agent should execute the current configuration or move to the next. The updating process is formally defined in Eq. 4.

$$\alpha_{t,i} = \sum_{\imath=i-1}^{i} \alpha_{t-1,\imath} \cdot \gamma_{t,i-\imath} \tag{4}$$

Using a set of rules to determine the value of the controller γ is not practical. For example, for the instruction "move to the table" or "move past the table", it is hard for an agent to decide whether to execute the current configuration or to move to the next one only based observing or not-observing the "table". To address this issue, we let the agent learn the value of γ based on three aspects of information. The first one is the previous hidden state h_{t-1}; the second one is the attended image representation $\bar{I}_t$ at the current step; the third one is the similarity score S_t between the landmark representations and the object representations, Eq. 5 shows how to get the similarity score S_t, and α_{t-1} is the attention weight at the previous step.

$$S_t = \tilde{C}_L \cdot O \cdot \alpha_{t-1} \tag{5}$$

Then, we use a fully connected layer to predict the distribution $\gamma_t = \text{FC}_\gamma\left(\left[h_{t-1}; \bar{I}_t; S_t\right]\right)$. Finally, we apply the state attention to $\tilde{C}$ to get the grounded instruction representation based on the configuration $\hat{C} = \sum_i \alpha_{t,i} \cdot \tilde{C}_i$, which is used as the language input to the decoder $C_t^* = \hat{C}$.

3.6 Visual Representation Alignment

The intuition to leverage the object representation is to select navigable images by aligning the object representation with the configuration representation. We use two levels of soft attention, first over the objects in each image by configuration representation $\hat{C}$, and second over all images guided by the previous context h_{t-1}.

$$\begin{aligned}\hat{O}_j &= \text{SoftAttn}_{\text{obj}}(Q = \hat{C}; K = O_j; V = O_j) \\ \hat{I} &= \text{SoftAttn}_{\text{objimg}}(Q = h_{t-1}; K = \hat{O}; V = I^{'})\end{aligned} \tag{6}$$

where $\hat{O} = \left[\hat{O}_1, \hat{O}_2, \cdots, \hat{O}_n\right]$. We use the image representation $\hat{I}$, that has aligned the objects with the configurations, as the visual input to the decoder $I_t^* = \hat{I}$.

3.7 Navigable Viewpoint Selection

We obtain a new decoder context h_t, as described in Section 3.2, with configuration input C_t^* and visual input I_t^*, where t is the current step. The next step is to predict the viewpoint with the image that has the highest correlation with the current context and configuration, calculated by $z_{t,j} = \left\langle I_j^{'}, \text{FC}_{\text{pred}}\left([C_t^*; h_t]\right)\right\rangle$, where $\text{FC}_{\text{pred}}(\cdot)$ is a fully-connected layer. We sum the scores of the three elevations for each navigable viewpoint k as $\zeta_{t,k} = \sum_{j \in \kappa_k} z_{t,j}$, where κ_k is the set of three elevations' image indexes. The predicted navigable viewpoint distribution p_t can be calculated with $p_t = \text{softmax}(\zeta_t)$.

3.8 Training and Inference

We train our model with two state-of-the-art training strategies in this task. (1) **T1**: We follow Self-Monitor (Ma et al., 2019) optimizing the model with a cross-entropy loss to maximize the likelihood of the ground-truth navigable viewpoint given by the model, and a mean squared error loss to minimize the normalized distance in units of length from the current viewpoint to the goal destination. At each step, the next viewpoint is selected by sampling the predicted probability of each navigable viewpoint. (2) **T2**: We follow (Tan et al., 2019) training the model with the mixture of Imitation Learning and Reinforcement Learning, where Imitation Learning minimizes the cross-entropy loss of the prediction and always samples the ground-truth navigable viewpoint at each time step, and Reinforcement Learning uses policy gradient to update the parameters of the model.

During inference, we conduct a greedy search with the highest probability of the next viewpoints to generate the trajectory. It should be noticed that beam search with a beam size greater than one is not practical because the agent needs to move forward and backward in the physical world, resulting in a long trail trajectory before making a decision.

4 Experimental Setup

Dataset We evaluate our model with Room-to-Room (R2R) dataset (Anderson et al., 2018), which is built upon the Matterport3D dataset (Chang et al.,

	Method	Validation-Seen			Validation-Unseen			Test(Unseen)		
		NE ↓	**SR** ↑	**SPL** ↑	NE ↓	**SR** ↑	**SPL** ↑	NE ↓	**SR** ↑	**SPL** ↑
1	Random (Anderson et al., 2018)	9.45	0.16	-	9.23	0.16	-	9.77	0.13	0.12
2	Student-forcing (Anderson et al., 2018)	6.01	0.39	-	7.81	0.22	-	7.85	0.20	0.18
3	Speaker-Follower (Fried et al., 2018)	4.36	0.54	-	7.22	0.27	-	-	-	-
4	Speaker-Follower*	3.66	0.66	0.58	6.62	0.36	-	6.62	0.35	0.28
5	Self-Monitor* (Ma et al., 2019)	**3.22**	**0.67**	0.58	5.52	0.45	0.32	**5.67**	0.48	0.35
6	Environment Dropout* (Tan et al., 2019)	4.19	0.58	0.55	**5.43**	**0.48**	**0.44**	-	**0.52**	**0.47**
7	Environment Dropout + BERT*	4.40	0.61	0.57	5.54	0.46	0.43	-	-	-
8	SpC-NAV*	4.09	0.65	**0.61**	5.92	0.45	0.42	6.22	0.46	0.44

Table 1: **Experimental Result comparing with baseline models.** * means data augmentation.

Method	Val-Seen			Val-Unseen			Test(Unseen)		
	NE ↓	**SR** ↑	**SPL** ↑	NE ↓	**SR** ↑	**SPL** ↑	NE ↓	**SR** ↑	**SPL** ↑
Self-Monitor (T1)	3.72	0.63	0.56	5.98	0.44	0.30	-	-	-
Sub-Instruction(T1)	-	-	-	**6.16**	**0.42**	**0.32**	-	-	-
SpC-NAV+T1	3.95	0.65	0.59	6.51	0.39	**0.32**	6.22	0.42	0.35
EnvDrop (T2)	4.71	0.55	0.53	**5.49**	**0.47**	**0.43**	-	-	-
Sub-Instruction(T2)	-	-	-	5.67	0.47	**0.43**	-	-	-
SpC-NAV+T2	**4.68**	**0.59**	**0.56**	6.68	0.44	0.39	6.25	0.45	0.43

Table 2: **Experimental Result with Different Training Strategies**. T1 and T2 are two training strategies.

2017). This dataset has 7,189 paths and 21,567 instructions with an average length of 29 words. The whole dataset is divided into training, seen validation, unseen validation, and (unseen) test sets. The seen validation set shares the same visual environments with the training set, while unseen validation and test sets contain different environments.

Evaluation Metrics We report three evaluation metrics. (1) Navigation Error (NE): the mean of the shortest path distance between the agent's final position and the goal location. (2) Success Rate (SR): the percentage of the cases where the predicted final position lays within 3m from the goal location. (3) Success rate weighted by normalized inverse Path Length (SPL): SPL normalize Success Rate by trajectory length (Anderson et al., 2018). SPL is recommended as the primary metric because it considers both the effectiveness and efficiency of navigation performance.

4.1 Baseline Models

We mainly compare Spc-NAV with the following baseline models. **Seq2Seq** (Anderson et al., 2018) trained an encoder-decoder model with two learning strategies of random and student-forcing. **Speaker-Follower** (Fried et al., 2018) introduced a speaker module to synthesize new instructions to train the follower module. **Self-Monitor** (Ma et al., 2019) co-grounded instructions and image based on soft attention mechanism. **Environmental Dropout** (Tan et al., 2019) proposed a neural agent trained with the method of the mixture of Imitation Learning and Reinforcement Learning. **Sub-instruction** (Hong et al., 2020) segmented the instruction into sub-instructions and designed a shifting attention module to ensure the sequential execution order between sub-instructions. The differences between Sub-instruction and our model has been discussed in Section 2.

4.2 Implementation Details

We implement SpC-NAV using PyTorch [4] We use 768-d BERT-base (Devlin et al., 2018) (frozen) as the embedding of the raw instruction, and get its 512-d contextual embedding by LSTM. We encode the representations of the motion indicator and the landmark in each configuration with 300-d GloVe embedding respectively, and concatenate them with the 512-d configuration representation to obtain the enriched configuration representation (1112-d). We use 300-d GloVe embedding of object label representation to calculate similarity score S with configuration representation. We trained an auto-encoder to map 2048-d object visual representation from Faster R-CNN to 152-d, and use it to obtain the attended object representation $\hat{O}$. We optimize using ADAM with learning rate $1e-4$ in batches of 64. We used a rule-based parser to obtain the spatial configuration and spatial semantic elements. This provides some noisy extractions. Appendix A.2 includes the details about the accuracy of the parser based on our manual annotations of a subset of instructions.

5 Results and Analysis

Table 1 shows the main performance metrics of our proposed SpC-NAV, compared with the baseline models on seen/unseen validation set and unseen testing set. To achieve the best result, SpC-NAV is trained with the training strategy T2 (see Section 3.8) and the data augmentation proposed in (Tan et al., 2019). Our model improves the performance in the seen environment and obtains com-

[4] https://github.com/zhangyuejoslin/SpC-NAV

petitive results in the unseen environment. Since we use BERT as the input to the encoder while the baseline models use basic word embeddings, we replace the word representations in Environment Dropout with BERT for a fair comparison. Although the richer language representations help the performance, our model still achieves better results, especially in the seen environments. It indicates that the spatial configuration and spatial elements indeed improve the agent's reasoning ability.

Training strategies are orthogonal to our work, and our model is friendly to the strategies widely used in the literature (T1/T2) (see Section 3.8). We evaluate SpC-NAV with both T1 and T2 and compare the results with their baseline models as well as Sub-Instruction. We do not apply data augmentation in this setting. As shown in Table 2, SpC-NAV achieves consistent improvement in the seen environment compared with all the baselines. In the unseen environment, training with T1, SpC-NAV outperforms Self-monitor (and is even comparable to it with data augmentation) and performs similarly as Sub-Instruction. However, training with T2, our model does not outperform Environment Dropout and Sub-Instruction in unseen environments. We analyze the errors in Section 5.2.

5.1 Ablation Analysis

Table 3 shows how various spatial semantic elements influence the performance of the model. The model is trained with the training strategy T1. Row#1 is our model without considering spatial elements. From row#2 to row#3, we incorporate the representations of the motion indicator and the landmark into spatial configuration representation incrementally. In row#4, we use the similarity score between the landmark representations in the configuration and the object label representations in the image to control the transitions between spatial configurations. All motion indicator, landmark and similarity score improve the performance. After applying the similarity score, the large gain indicates that the connection between landmarks and objects is important in language grounding.

5.2 Qualitative Analysis

Seen Environment

We analyze some qualitative examples to find out how the spatial semantics improve the model. For the semantics of motion, we find that our model can improve the cases that motions contain "up" and "down" after adding the representation of motion indicator. Figure 3 (a) shows an example of such a scenario. The spatial configuration is "walk up the stairs", and the agent could find the right viewpoints after we incorporated the representation of the motion indicator "walk up". However, the model makes more mistakes in the cases that the motion indicators are highly related to the objects, such as "walk through", "walk past", and "walk towards", which need the landmark information. In these latter cases, the model should consider both motions and landmarks together. In another experiment, we added the landmark representation. Figure 3 (b) shows an example that the spatial configurations is "walk past the dining room table". The agent can select the correct viewpoints when we incorporate the representation of landmark "dining room table". We also analyze the influence of the similarity score, and found that when the information in the current configuration is not sufficient to make a decision, the similarity score will assist in choosing the next configuration. For example, in Figure 3 (c), the spatial configurations are "turn right" and "walk past the couch". Without using the similarity score in controlling the transitions between configurations, the agent tends to select a viewpoint in the "right" direction. But with similarity score, the agent will consider both "turn right" and "walk past the couch", and selects the correct viewpoint that the "couch" can be seen.

		Validation-Seen			Validation-Unseen		
	Model	NE↓	**SR↑**	**SPL↑**	NE↓	**SR↑**	**SPL↑**
1	SpC-NAV	4.11	0.62	0.53	6.49	0.39	0.29
2	SpC-NAV$_{\text{M}}$	**3.88**	0.62	0.53	**6.21**	**0.40**	0.28
3	SpC-NAV$_{\text{M+L}}$	4.01	0.62	0.54	6.27	0.39	0.29
4	SpC-NAV$_{\text{M+L+S}}$	3.95	**0.65**	**0.59**	6.51	0.39	**0.32**

Table 3: **Ablation study with different spatial semantics.** The subscription letters mean the model took those information into account; *M: motion indicator*; *L: landmark*; *S: similarity score*.

Unseen Environment

Table 1 and Table 2 show that our model does not outperform Environment Dropout in the unseen environments. We noticed that the main error is that some objects can not be detected in the image by the object detection model. This is more problematic for our model because we explicitly align the landmark phrases with the detected objects. For example, in Fig 4 (a), the agent selects the correct viewpoint when the configuration is "Walk to the glass door" because the connection between the landmark "glass door" and the object "door" has

(a) ±Motion Indicator
Walk up the stairs.

(b) ±Landmark
Walk past the dining room table.

(c) ±Similarity Score
Turn right, and walk past the couch.

Figure 3: **Analysis of Seen examples.** In these three scenarios, the corresponding spatial configurations are provided. Green arrows in the above figures show the correct trajectory was selected after the additional spatial semantics; red arrows show without that information the agent went wrong.

(a) Walk to the **glass door**.

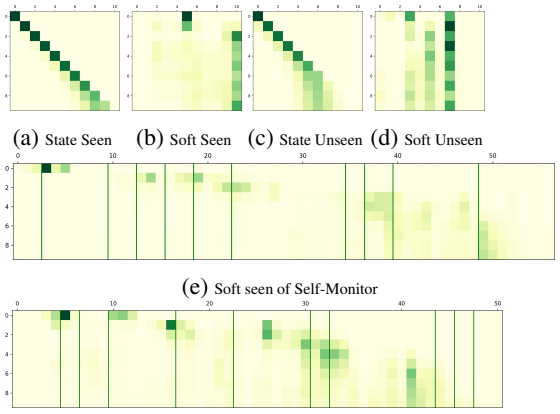
(b) Go to the **pottery**.

Figure 4: **Analysis of an Unseen example**

been learned in training set. In Fig (b), the agent is wrong when the configuration is "Go to the pottery." because the "pottery" is not detected at the initial perspective and the word "pottery" never appears in the training set. However, the agent selects a viewpoint that a bounding box contains a pottery. The gap between seen and unseen become larger after data augmentation since our model is able to capture the structure of the language by observing more examples. It can deal with the variations in the instructions and improve the performance in the seen environment, but it fails to deal with the novel objects and visual variations in the unseen environments. This is an orthogonal issue addressed in zero-shot learning (Blukis et al., 2020).

5.3 State Attention Visualization

We visualize the state attention and the soft attention weights over configurations. As shown in Fig 5a and Fig 5c, our designed state attention demonstrates that the grounded configuration shifts gradually from the first configuration to the last in both seen and unseen environments. We apply the soft attention used in Self-Monitor on spatial configurations, as shown in Fig 5b and Fig 5d, it can not preserve the sequential execution order. We also show the soft attention weights of the grounded instruction in the Self-Monitor by splitting the instructions with the boundaries of our configurations. As shown in Fig 5e and Fig 5f, although their attention weights show the gradual shift, many configurations are skipped.



(a) State Seen (b) Soft Seen (c) State Unseen (d) Soft Unseen

(e) Soft seen of Self-Monitor

(f) Soft unseen of Self-Monitor

Figure 5: **Attention weights of various attention strategies with seen and unseen examples.** The horizontal axis is the configuration order, and the vertical axis is the temporal order of the steps taken by the agent. Each row in sub-figures show the attention distribution over the configurations (or tokens) in an instruction at each time step. The green vertical lines in Figure (e) and Figure (f) indicate the split points of the configurations in the instruction.

6 Conclusion

We propose a neural agent that incorporates the semantic elements of spatial language for vision-and-language navigation. We use the notion of spatial configurations as the main linguistic unit of the instructions and enhance the spatial configuration representation with the representations of motion indicator and landmark. We design a state attention to guarantee the sequential execution order of configurations and use the similarity score between the representations of landmarks and objects to control the transitions between configurations. Based on our results, incorporating the spatial semantics improves reasoning ability over navigation. Future work could investigate more fine-grained spatial semantics and the geometry of spatial relations. Also, we will deal with novel objects in a zero-shot setting to improve the unseen environments results.

Acknowledgement

This project is supported by National Science Foundation (NSF) CAREER award 2028626 and partially supported by the Office of Naval Research (ONR) grant N00014-20-1-2005. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation nor the Office of Naval Research. We thank all reviewers for their thoughtful comments and suggestions.

References

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. 2018. 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.

Valts Blukis, Ross A Knepper, and Yoav Artzi. 2020. Few-shot object grounding and mapping for natural language robot instruction following. *arXiv preprint arXiv:2011.07384*.

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. 2017. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*.

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

Soham Dan, Parisa Kordjamshidi, Julia Bonn, Archna Bhatia, Zheng Cai, Martha Palmer, and Dan Roth. 2020. From spatial relations to spatial configurations. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 5855–5864.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. 2018. Speaker-follower models for vision-and-language navigation. In *Advances in Neural Information Processing Systems*, pages 3314–3325.

Mehdi Ghanimifard and Simon Dobnik. 2019. What goes into a word: generating image descriptions with top-down spatial knowledge. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 540–551.

Yicong Hong, Cristian Rodriguez-Opazo, Qi Wu, and Stephen Gould. 2020. Sub-instruction aware vision-and-language navigation. *arXiv preprint arXiv:2004.02707*.

Ronghang Hu, Daniel Fried, Anna Rohrbach, Dan Klein, Kate Saenko, et al. 2019. Are you looking? grounding to multiple modalities in vision-and-language navigation. *arXiv preprint arXiv:1906.00347*.

Thomas Kollar, Stefanie Tellex, Deb Roy, and Nicholas Roy. 2010. Toward understanding natural language directions. In *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 259–266. IEEE.

Chih-Yao Ma, Jiasen Lu, Zuxuan Wu, Ghassan AlRegib, Zsolt Kira, Richard Socher, and Caiming Xiong. 2019. Self-monitoring navigation agent via auxiliary progress estimation. *arXiv preprint arXiv:1901.03035*.

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

Hongyuan Mei, Mohit Bansal, and Matthew R Walter. 2016. Listen, attend, and walk: Neural mapping of navigational instructions to action sequences. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Terry Regier. 1996. *The human semantic potential: Spatial language and constrained connectionism*. MIT Press.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Hao Tan, Licheng Yu, and Mohit Bansal. 2019. Learning to navigate unseen environments: Back translation with environmental dropout. *arXiv preprint arXiv:1904.04195*.

Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R Walter, Ashis Gopal Banerjee, Seth Teller, and Nicholas Roy. 2011. Understanding natural language commands for robotic navigation and mobile manipulation. In *Twenty-fifth AAAI conference on artificial intelligence*.

Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. 2019. Vision-and-dialog navigation. *arXiv preprint arXiv:1907.04957*.

Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. 2019. 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.

Haonan Yu, Xiaochen Lian, Haichao Zhang, and Wei Xu. 2018. Guided feature transformation (gft): A neural language grounding module for embodied agents. *arXiv preprint arXiv:1805.08329*.

Chen Zheng and Parisa Kordjamshidi. 2020. Srlgrn: Semantic role labeling graph reasoning network. *arXiv preprint arXiv:2010.03604*.

Wang Zhu, Hexiang Hu, Jiacheng Chen, Zhiwei Deng, Vihan Jain, Eugene Ie, and Fei Sha. 2020. Babywalk: Going farther in vision-and-language navigation by taking baby steps. *arXiv preprint arXiv:2005.04625*.

A Appendix

A.1 Visual Representation Analysis

In this section, we experiment with three types of object representations introduced in Section 3.6, which are object label representation and object visual representation and the combination of these two types of object representation. As shown in Table 4, object visual representation performs better in unseen environments, and we use it to get attended object representation $\hat{O}$ in our best model. This experiment does not consider the similarity score between the representations of landmarks and objects.

	Validation-Seen			Validation-Unseen		
Repr.	NE↓	**SR**↑	**SPL**↑	NE↓	**SR**↑	**SPL**↑
Label	4.51	0.58	0.52	6.43	0.37	0.28
Visual	4.01	0.62	0.54	6.27	0.39	0.29
Label + Visual	4.45	0.59	0.53	6.54	0.37	0.28

Table 4: Result with Different Visual Representations.

A.2 Parsing Analysis

The performance of our rule-based parser influences the result of navigation. To evaluate it, we manually annotated 845 spatial configurations for 200 instructions. We annotated motion indicators, spatial indicators and landmarks in those configurations. Our parser achieves an accuracy of 85% in extracting the spatial configurations. For the extraction of spatial elements, the accuracy is 73% for motion/spatial indicators, and 77% for landmarks.

In the following, we analyze two types of error in getting spatial configurations (Split Error and Order Error), and other errors that generated in the extraction of motion indicator, spatial indicator and landmark.

Split Error

The split configuration may only convey the spatial position of objects rather than executable navigation information. For example, in the instruction, "Turn left. There is a rocking chair in it," two configurations are generated based on our split method: "Turn left" and "There is a rocking chair in it." However, the second configuration is not an independent spatial configuration because it indicates no motion, and it is attached to the previous configuration.

Order Error

We order the configurations based on their occurrence in the sentence. However, there are cases that the configurations have an inverted order. For instance, "Stop once you pass the counter on the right" is split as "stop" and "you pass the counter on the right." However, the implied sequence is inverted because of "once".

Motion Indicator and Spatial Indicator

We build a vocabulary based on training data to collect the commonly used verb phrases, and the vocabulary size is 241. Table 5 shows some examples. If the motion indicator and spatial indicator does not show in the vocabulary, we will treat the verbs as the motion indicators and prepositions as spatial indicators in configurations. With this method, we can get 73% accuracy since there are expressions that never appear in the training dataset, and it is hard to extract the complete verb phrases only based on pos-tag.

Landmark

We extract the noun phrases of each configuration as landmark and can get 77% accuracy. However, there are some special cases, for example, "a left" in "make a left" is extracted as noun chunk, but it can not be treated as a landmark. Also, for the expression "middle of the doorway", "the middle" and "the doorway" are both noun chunks, but the whole phrase is the landmark instead of separated ones.

head straight, walk through, walk down, walk into, walk inside, turn around, turn left, make a left turn, jump over, move forward, turn slightly right

Table 5: Verb Phrases Examples