# Explicit Object Relation Alignment for Vision and Language Navigation

Anonymous ACL submission

## Abstract

We propose a neural agent to solve the navigation instruction following problem in a photo-realistic environment. We explicitly align the spatial information in both instruction and the visual environment, including landmarks and spatial relationships between the agent and landmarks. Our method significantly improves the baseline and is competitive with the SOTA in unseen environments. The qualitative analysis shows that explicitly modeled spatial reasoning improves the explainability of the action decisions and the generalizability of the model.

## 1 Introduction

Vision and Language Navigation (VLN) task (Anderson et al., 2018) requires the agent to carry out a sequence of actions in an indoor photo-realistic simulated environment in response to corresponding natural language instructions, as shown in figure 1. It is a challenging task because, apart from understanding the language and vision modalities, the agent needs to learn the connection between them without explicit intermediate supervision.

To address this challenge, recent works start to consider the semantic structure from both language and vision sides. Hong et al. (2020a) train an implicit entity- relationship graph allowing an agent to learn the latent concepts and relationships between different components (scene, object and direction). They use the object features extracted from Faster-RCNN (Ren et al., 2015) instead of only using ResNet visual features, which can easily overfit to the training environment (Hu et al., 2019). Although the grounding ability of their agent improves, their experimental results show that the object features do not help the navigation independently unless their relationships to the scene and direction are modeled. And we are left with the question of how to achieve successful navigation with object representations independently.

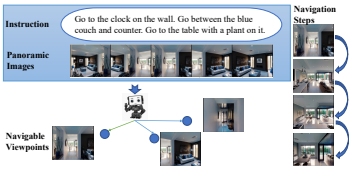



Figure 1: **VLN Task Demonstration.** The agent generates a navigation trajectory composed of navigable viewpoints selected based on the given instruction and the panoramic images at each step. The green arrow shows the ground-truth navigable viewpoint.

Besides, the recent research finds that indoor navigation agents rely on both landmark and direction tokens in the instruction when making action decisions (Zhu et al., 2021). To model landmarks, one of the difficulties is letting the agent know which landmarks it should pay attention to at each navigation step. Previous works (Tan et al., 2019; Ma et al., 2018; Wang et al., 2019; Zhu et al., 2020) mainly use the surrounding visual information as a clue to indicate the landmark tokens that the agent should focus on. However, the semantics of instruction should also play an important role. For example, with the understanding of the instruction "go to the table with chair, and then walk towards the door", the agent needs to give the same attention to "table" and "chair", and less attention to "door" at the first navigation step. In terms of direction tokens, no method distinguishes the direction tokens related to motions, such as "turn left", and the spatial description of landmarks, such as "table on the left". However, modeling these different cases explicitly can help explain the agent's actions.

In this paper, we propose a neural agent, namely *Explicit Object Relation Alignment Agent* (EXOR), to explicitly align the spatial semantics between instructions and the visual environment. Specifically, we first select the important landmarks in the instructions after splitting the long instruction into spatial configurations (Dan et al., 2020; Zhang et al., 2021). Then we obtain the most relevant objects in the visual environment based on their align-

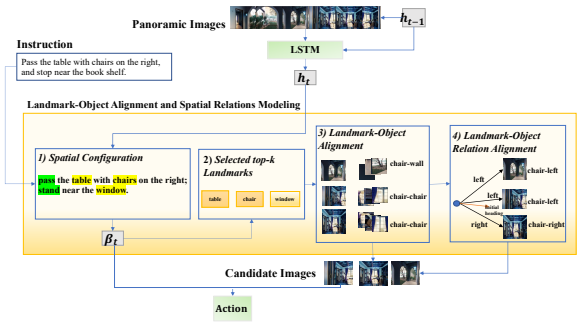



Figure 2: **Model Architecture.**

ments with the selected landmarks and use them to enrich the image representation further. Besides, we map the encoding of spatial relations between the agent and landmarks in the instruction and the encoding of the agent's perspective based on its angle with the images. None of the previous work modeled the explicit spatial relations considering the agent's perspective for this task.

Our contribution is summarized as follows. 1. Our agent learns to focus on the visual objects conditioned on the landmarks in the instructions based on their learnt explicit alignments. 2. We model spatial relations between the agent and landmarks from both instruction and visual environments to enhance their alignments. 3. Our proposed method improves the strong baseline and is competitive with SOTA in the unseen environment; it improves the spatial reasoning ability and explainability.

## 2 Related Work

The visual and textual co-grounding in the VLN task is to learn the connection between instruction and the visual environment. The early methods (Anderson et al., 2018; Ma et al., 2018; Tan et al., 2019; Wang et al., 2019) use attention mechanisms to build language and vision connections in neural navigation agents. The second branch of works (Hu et al., 2019; Hao et al., 2020; Majumdar et al., 2020; Hong et al., 2020a) obtains the pre-trained vision and language representation based on the transformer models to improve the navigation performance largely. The third branch of works (Hong et al., 2020b; Li et al., 2021; Qi et al., 2020; Zhang et al., 2021) models the semantic structure from both language and vision sides. In this paper, we mainly compare with the third branch of works.

## 3 Method

**Base Model** Our model is built upon Environment Dropout Agent (Tan et al., 2019), which uses an LSTM-based sequence-to-sequence architecture. In the base model, the language representation $s$ is obtained with an LSTM encoder. The image representations are the concatenation of the ResNet visual features and direction encoding. Formally, the panoramic image features and candidate image features are represented as $f^p$ and $f^c$ respectively. The agent first attends to the panoramic image representation $f^p$ with the previously hidden context feature $h_{t-1}$ of the LSTM decoder. The attended panoramic image features are input to the LSTM decoder to get the agent's current state representation $h_t$. The agent then uses $h_t$ to attend to the instructions and makes action decisions by learning the connections between the weighted instruction and candidate images. As shown in figure 2, our method is to model the alignments between landmarks and objects and their spatial relations to enrich the image features of the base model.

**Landmark-object alignment and spatial relations modeling** We describe four components of this module as follows.

*1) Spatial Configuration Representation* We split the long instructions into smaller sub-instructions called spatial configurations. A spatial configuration contains fine-grained spatial roles, such as motion indicator, landmark, spatial indicator, trajector (Dan et al., 2020). For example, the instruction "go to the bathroom and stop" can be split into two spatial configurations, which are "go to the bathroom" and "stop". In the first configuration, "go" is the motion indicator; "bathroom" is the landmark. In the second configuration, "stop" is the motion indicator. We follow the method used in (Zhang et al., 2021) to re-organize the contextual embedding of tokens $s$ into $m$ spatial configuration representations $C = [C_1, C_2 \dots C_m]$. The hidden context $h_t$ of the decoder then attends to the spatial configurations $C$ to obtain the attended spatial configuration weights denoted as $\beta = softmax(C^T W_c h_t)$, where $W_c$ is the learned weights.

*2) Landmark Selection* Landmark phrases in instructions are split into groups per spatial configuration. We assign the attention weights of each spatial configuration to all its included landmarks. The attention weights of landmarks are the same if they are in the same configuration. Then we sort all weighted landmarks and select the top-$k$ important ones for the agent to focus on at each navigation step. Formally, each configuration contains $n$ landmarks, denoted as $L =< L_1, L_2, \cdots, L_n >$. The total number of landmarks is $m * n$ in $m$ spatial

configurations. After sorting all landmarks based on the spatial configuration weights $\beta$, we can obtain top-$k$ selected landmark representations, as $\tilde{L} =< \tilde{L}_1, \tilde{L}_2, \cdots, \tilde{L}_k >$. We get the best result when $k$ is 3 (see Appendix A.3 for the experiment).

*3) Landmark-Object Alignment* After getting top-$k$ landmarks, the next step is to align them with the corresponding objects in the image. We use Faster-RCNN to detect 36 objects in each image, and the object representation of the i-th image is $O_i = [o_{i,1}, o_{i,2}, \cdots, o_{i,36}]$. We compute the cosine similarity scores between the j-th landmark in top-$k$ landmarks and all objects in the i-th image, and select the object with the highest similarity score as the most relevant object to the j-th landmark, as $\hat{O}_{i,L_j} = max(cos\_sim(\tilde{L}_j, O_i))$. Then the aligned objects in the i-th image are denoted $\hat{O}_i = [\hat{O}_{i,L_1}, \hat{O}_{i,L_2}, \cdots, \hat{O}_{i,L_k}]$. We get $k$ aligned objects since we have top-$k$ landmarks. Finally, we concatenate the aligned object representations with the candidate image features $f^c$, and the i-th candidate image feature is updated as $\hat{f}_i^c = [f_i^c; \hat{O}_i^c]$.

*4) Landmark-Object Relation Alignment* On the text side, there are mainly three different cases of spatial relations used in the navigation instructions. Case 1. Motions verbs, such as "turn left to the table"; Case 2. Relative spatial relationships between agent and landmarks, such as "table on your left"; Case 3. Spatial relationships between landmarks, such as "vase on the table". This work mainly investigates the spatial relations from the agent's perspective, and we only model the first two cases. We extract "landmark-relation" pairs for each landmark in the instructions (based on syntactic rules). For Case 1, we pair the spatial relation with all landmarks in the configuration. For example, "turn left to the table with chair", the extracted pairs are {table-left} and {chair-left}. For Case 2, we pair the relation with the related landmark. For example, "go to the sofa on the right.", the extracted pair is {sofa-right}. We encode the spatial relations for the landmarks in six bits $[left, right, front, back, up, down]$. The bit is set to 1 for the landmark if its paired relation has the corresponding value. On the image side, we encode the six spatial relations too. We obtain the spatial relations of objects in the visual environment based on the relative angle, the difference between the agent's initial direction and the navigable direction. The spatial relations are the same for all objects if they are in the same image.

Formally, for the obtained top-$k$ landmarks, we denote their spatial encoding as $R^{\hat{L}} = [R_1^{\hat{L}}, R_2^{\hat{L}}, \cdots, R_k^{\hat{L}}]$. For the top-$k$ objects aligned with those landmarks, the spatial relations in i-th navigable image are represented as $R_i^{\hat{O}} = [R_{i,1}^{\hat{O}}, R_{i,2}^{\hat{O}}, \cdots, R_{i,k}^{\hat{O}}]$. We compute the inner product of the spatial encoding between top-$k$ landmarks and the top-$k$ aligned objects to obtain the spatial similarity score between the instruction and the i-th image, that is, $sim_i^R = R^{\hat{L}} \cdot R_i^{\hat{O}}$. Then we concatenate each aligned object spatial encoding with the corresponding similarity score, denoted as $\hat{O}_{i,R} = [[R_{i,1}^{\hat{O}}; sim_{i,1}^R], [R_{i,2}^{\hat{O}}; sim_{i,2}^R], \cdots, [R_{i,k}^{\hat{O}}; sim_{i,k}^R]]$. Finally, we further concatenate $\hat{O}_{i,R}$ with the candidate image features $\hat{f}_i^c$ which is concatenated with the aligned object features , and i-th candidate images features is updated as $\hat{\hat{f}}_i^c = [\hat{f}_i^c; \hat{O}_{i,g}]$. The updated image representations are then used to make action decisions for the agent.

## 4 Experiments

**Dataset** We use Room-Room(R2R) dataset (Anderson et al., 2018) built upon the Matterport3D dataset. It contains 7198 paths and 21567 instructions with an average length of 29 words. The whole dataset is divided into training, seen validation, unseen validation, and unseen test set.

**Evaluation Metrics** We mainly report three evaluation metrics. Success Rate (SR), Success rate weighted by normalized inverse Path Length (SPL) (Anderson et al., 2018), and the Success weighted by normalized Dynamic Time Warping (SDTW) (Ilharco et al., 2019). Appendix A.1 shows their detailed description.

**Results and Analysis** Table 1 shows the performance of our model compared with baselines and other competitive models on unseen validation and test set. Our result is better than the baseline model even with their augmented data (Tan et al., 2019) (Row#1 and Row#2), showing our improved generalizability. We obtain significantly better results than SpC-NAV, which models the semantic structure in both language and image modalities. Compared with OAAM, which learns the object-vision matching with the augmented data, we get much better SDTW, showing that our agent can genuinely follow the instruction to the destination. However, Ent-Rel (SOTA) achieves better results, for which we provide further analysis in the next section.

| | Method | Val Unseen SR ↑ | Val Unseen SPL↑ | Val Unseen SDTW↑ | Test(Unseen) SR ↑ | Test(Unseen) SPL ↑ |
|---|---|---|---|---|---|---|
| 1 | Env-Drop (Tan et al., 2019) | 0.47 | 0.43 | - | - | - |
| 2 | Env-Drop* | 0.50 | 0.48 | 0.37 | 0.50 | 0.47 |
| 3 | SpC-NAV (Zhang et al., 2021) | 0.45 | 0.42 | - | 0.46 | 0.44 |
| 4 | OAAM* (Qi et al., 2020) | 0.54 | 0.50 | 0.39 | 0.53 | 0.50 |
| 5 | Ent-Rel (Hong et al., 2020a) | 0.52 | 0.50 | 0.46 | 0.51 | 0.48 |
| 6 | EXOR (ours) | 0.52 | 0.49 | 0.46 | 0.49 | 0.46 |

Table 1: Experimental Results Comparing with Baseline Models (* means data augmentation).

| | | Ent-Rel SR↑ | Ent-Rel SPL↑ | EXOR(ours) SR↑ | EXOR(ours) SPL↑ |
|---|---|---|---|---|---|
| 1 | Mask Scene | 0.47 | 0.44 | 0.48 | 0.46 |
| 2 | No Mask | 0.52 | 0.50 | 0.50 | 0.48 |

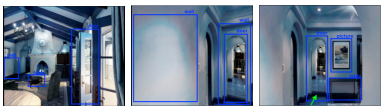
Table 2: Results on Scene & Object Alignment.

| | Method | Val Seen SR↑ | Val Seen SPL↑ | Val Seen SDTW↑ | Val Unseen SR↑ | Val Unseen SPL↑ | Val Unseen SDTW↑ |
|---|---|---|---|---|---|---|---|
| 1 | Baseline | 0.55 | 0.53 | 0.49 | 0.47 | 0.43 | 0.37 |
| 2 | Obj | 0.59 | 0.55 | 0.52 | 0.50 | 0.48 | 0.43 |
| 3 | Obj+Rel | **0.60** | **0.58** | **0.53** | **0.52** | **0.49** | **0.46** |
| 4 | Obj+Rel_v | 0.59 | 0.56 | 0.52 | **0.52** | 0.47 | 0.44 |

Table 3: Ablation Study.

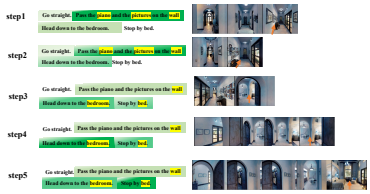
**Scene & Object Alignment** Ent-Rel(Hong et al., 2020a) distinguishes the landmarks which are *scene*s from *object*s. Scene tokens describe the location at a coarse level, such as “bathroom", while object tokens describe the exact landmarks, such as “table". To evaluate the agent’s performance given the instructions with only object tokens, we mask all scene tokens in the instructions and experiment on Ent-Rel and our model. Table 2 shows the experimental results in the unseen validation set. Compared with Ent-Rel, our model performs slightly better given the instruction with only object tokens but worse with scene and object tokens. One of the reasons is that Faster-RCNN often does not detect the scenes. For example, the aligned object labels in the image for the landmark "bedroom" are “floor", “roof", “wall", which are only parts of the bedroom. Our explicit modeling of the alignment between landmarks and objects can be easily applied to other VLN neural agents to enrich the visual representation. For Ent-Rel, our method not only can enrich their visual features, but the explicitly extracted spatial relations can also reduce the redundancy of their built entity relation graph. Potentially, our method can be helpful to improve the performance and explainability of their model.

**Ablation Study** Table 3 shows the ablation study results. Row#1 is the baseline model. Row#2 (*Obj*) shows that explicitly modeling important landmarks and aligned objects improve the performance compared to the baseline. *Rel* (row#3) is the result after modeling the spatial relation tokens describing the relative relation between agent and landmark. *Rel_v* (row#4) is the result after modeling the spatial relations in motions. The improved SDTW shows the modeling of spatial relations can help the agent to follow the instructions. However, the spatial terms directly describing the landmark are more helpful than the spatial terms in motions.



(a) Enter the “door” to the small “table” with a “painting” above.
v1: [door-door; table-table; painting-wall]
v2: [door-door; table-wall; painting-wall]
v3: [door-door; table-table; painting-picture]

(b) Head towards the “doors” on the left towards “kitchen”.
v1:left; v2:right; v3:right




(c) The green boxes are spatial configurations; darker green means higher weights; yellow boxes are the selected landmarks; the orange arrows are the path.

Figure 3: **Qualitative Examples.** Blue bounding boxes are the aligned objects. Green arrow is the selected correct viewpoint. v is the viewpoint, the alignment between landmarks and objects is [landmark-object].

**Qualitative Analysis** Figure 3 shows qualitative analysis examples. The selected k-important landmarks are “door”, “table”, “painting” in figure 3a. The agent makes a correct decision by selecting the viewpoint that contains the objects aligned with all three landmarks. Figure 3b shows an example after modeling spatial relations. Although three navigable viewpoints have the object "door", the agent selects the aligned object with the “left” direction. However, we find that relation alignments will be helpful when the object alignments are done correctly. Appendix A.4 shows the extra analysis. Also, in figure 3c, we provide an example to visualize the navigation process using the selected landmark based on the spatial configurations.

## 5 Conclusion

In this paper, we select the important landmarks from the linguistic instructions and design a neural model to let the agent focus on the aligned objects with the important landmarks. We also explicitly model the spatial relations between the agent and the landmarks from the agent’s perspective on both instruction and image sides. Our experiments show that both explicit object-landmark alignments and the spatial relations modeling improve the results.

## 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.

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.

Weituo Hao, Chunyuan Li, Xiujun Li, Lawrence Carin, and Jianfeng Gao. 2020. 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.

Yicong Hong, Cristian Rodriguez, Yuankai Qi, Qi Wu, and Stephen Gould. 2020a. Language and visual entity relationship graph for agent navigation. *Advances in Neural Information Processing Systems*, 33:7685–7696.

Yicong Hong, Cristian Rodriguez, Qi Wu, and Stephen Gould. 2020b. Sub-instruction aware vision-and-language navigation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3360–3376.

Ronghang Hu, Daniel Fried, Anna Rohrbach, Dan Klein, Trevor Darrell, and Kate Saenko. 2019. Are you looking? grounding to multiple modalities in vision-and-language navigation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6551–6557.

Gabriel Ilharco, Vihan Jain, Alexander Ku, Eugene Ie, and Jason Baldridge. 2019. General evaluation for instruction conditioned navigation using dynamic time warping. *arXiv preprint arXiv:1907.05446*.

Jialu Li, Hao Tan, and Mohit Bansal. 2021. Improving cross-modal alignment in vision language navigation via syntactic information. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1041–1050.

Chih-Yao Ma, Jiasen Lu, Zuxuan Wu, Ghassan AlRegib, Zsolt Kira, Richard Socher, and Caiming Xiong. 2018. Self-monitoring navigation agent via auxiliary progress estimation.

Arjun Majumdar, Ayush Shrivastava, Stefan Lee, Peter Anderson, Devi Parikh, and Dhruv Batra. 2020. Improving vision-and-language navigation with image-text pairs from the web. In *European Conference on Computer Vision*, pages 259–274. Springer.

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.

Yuankai Qi, Zizheng Pan, Shengping Zhang, Anton van den Hengel, and Qi Wu. 2020. Object-and-action aware model for visual language navigation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, pages 303–317. Springer.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99.

Hao Tan, Licheng Yu, and Mohit Bansal. 2019. Learning to navigate unseen environments: Back translation with environmental dropout. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2610–2621.

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/CVF Conference on Computer Vision and Pattern Recognition*, pages 6629–6638.

Yue Zhang, Quan Guo, and Parisa Kordjamshidi. 2021. Towards navigation by reasoning over spatial configurations. *arXiv preprint arXiv:2105.06839*.

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. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2539–2556.

Wanrong Zhu, Yuankai Qi, Pradyumna Narayana, Kazoo Sone, Sugato Basu, Xin Eric Wang, Qi Wu, Miguel Eckstein, and William Yang Wang. 2021. Diagnosing vision-and-language navigation: What really matters. *arXiv preprint arXiv:2103.16561*.

# A Appendix

## A.1 Evaluation Metric

We mainly report three evaluation metrics. (1) Success Rate (SR): the percentage of the cases where the predicted final position lays within 3m from the goal location. (2) Success rate weighted by normalized inverse Path Length (SPL) (Anderson et al., 2018): normalizes Success Rate by trajectory length. It considers both the effectiveness and efficiency of navigation performance. (3) the Success weighted by normalized Dynamic Time Warping (SDTW) (Ilharco et al., 2019): penalizes deviations from the referenced path and also considers the success rate.

## A.2 Implementation Details

We use PyTorch to implement our model. The contextual embedding is 512-d. We use 300-d GloVe (Pennington et al., 2014) embedding to represent motion indicator, landmark, and object label. The optimizer is ADAM, and the learning rate is $1e-4$ with a batch size of 32.

## A.3 The Number of Selected Landmarks

We experimented to find the best number of important landmarks the agent should select. Figure 4 shows the SPL results with different k values on validation seen and unseen dataset. We find that the best result is obtained when k is 3. It also shows that letting the agent focus on only one landmark or all landmarks in the instruction will hurt their navigation performance. Table4 shows the statistics on the extracted spatial configurations on train and validation seen/unseen dataset. On average, each instruction can be split into about four spatial configurations, and about 76% of spatial configurations contain landmarks. If so, selecting top3 landmarks means that the agent mainly focuses on the landmark-object alignment in 3 spatial configurations at most at each navigation step.

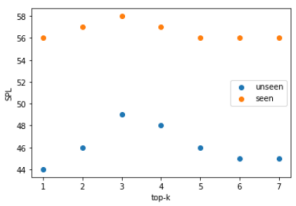



Figure 4: **SPL Results with Different K Values.**

| | | Train | Val Seen | Val Unseen |
|---|---|---|---|---|
| 1 | Instructions | 14025 | 1021 | 2349 |
| 2 | Configs | 58277 | 4301 | 9625 |
| 3 | Configs with Landmark | 44053 | 3225 | 7303 |
| 4 | Configs with relation | 13543 | 1142 | 2566 |

Table 4: Statistics of Spatial Configuration

(a) Walk past the "kitchen" towards the "dining room". Stop before you reach the "table".
v1: [kitchen-room; dining room-room; table-table]
v2: [kitchen-kitchen; dining room-room; table-kitchen]

(b) Turn right toward "bathroom". Stop at the top of the steps.
v1:left; v2:right;

Figure 5: **Extra Qualitative Examples**

## A.4 Extra Qualitative Examples

Figure 5a shows another example of landmark and object alignments. It contains two spatial configurations: "walk past the kitchen towards the dining room" and "stop before you reach the table". In the first configuration, the landmarks are "kitchen' and "dining room"; in the second configuration, the landmark is "table". By merely using the visual environment as a clue for viewpoint selection, the agent will select the second navigable viewpoint because of its detected "kitchen" view. However, based on the instruction semantics, the "kitchen" is an object the agent passes by, and the "table" is the final goal. In some cases, our method can handle such situations by using the selected landmarks. In this example, the model allows the agent to focus on the aligned object such as "table", which appear later in the spatial configuration. It increases the probability of selecting the first viewpoint. Also, we find that relation alignments modeling will be helpful only when the object alignments are done correctly. If the object alignments fail, for example, when the agent makes mistakes during navigation or the aligned objects can not be detected, modeling relations can worsen the situation. For instance, in figure 5b, for both navigable viewpoints, the object "bathroom" can not be detected, and in this case, further modeling relations leads to making wrong decisions.