Appendix

In this appendix, we first present additional details for the collection of GEL-R2R in Sec. A. And then we provide the implementation details of the pre-trained model in Sec. B. Finally, we compare several qualitative examples of our GELA model and the HAMT [3] baseline in Sec. C.

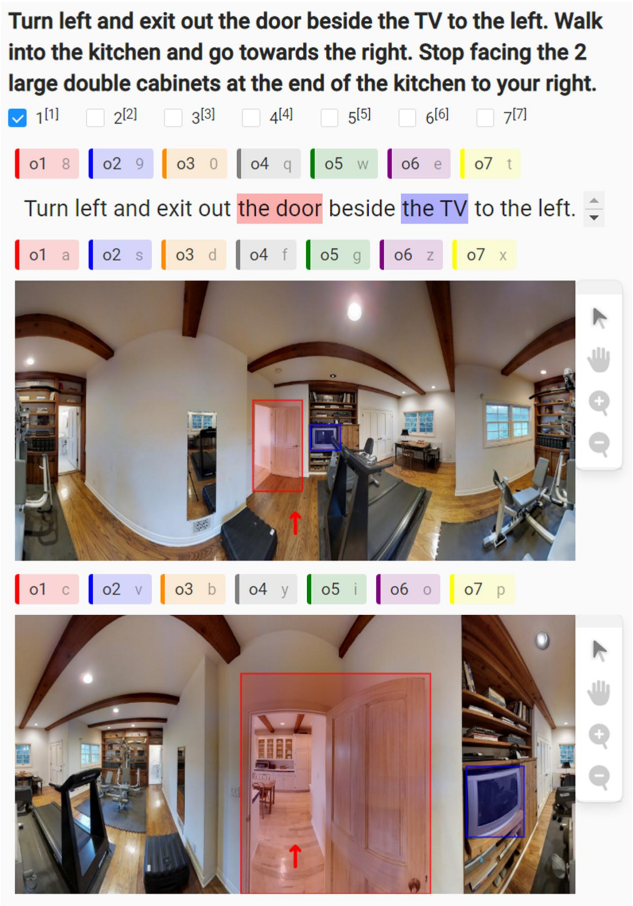
A. Data Collection Pipeline

A.1. Raw Data Preparation.

Our grounded entity-landmark annotations are based on the Room-to-Room (R2R) [1] dataset. For each navigation trajectory from R2R, we collect a sequence of 360-degree panoramas with an image size of 2048×1024 from Matterport3D simulator [2]. Additionally, we employ two skills in the preparation of raw data to increase the effectiveness and efficiency of entity-landmark grounding annotation. The first skill is turning each panorama so that the direction of the subsequent action heading is in the center of the image and marking the direction using the red arrow, as shown in Figure 1. Due to the fact that most landmarks are located near the direction of the next action, this skill might improve the speed with which the annotators identify the landmarks and reduce the phenomenon whereby landmarks are divided by the edge. The second skill is utilizing alignment information of sub-instructions and sub-trajectories modified from FG-R2R [8], where a sub-instruction may sometimes incorrectly match a viewpoint rather than a sub-trajectory. As presented in Figure 1, "Turn left and exit out the door beside the TV to the left" is the first sub-instruction of the overall instruction, and the following two panoramas are the visual observations of the agent at two viewpoints of the corresponding sub-trajectory. As a result, rather than having to search through every panorama along the path, annotators only need to find effective landmarks in the several corresponding panoramas.

A.2. Annotation Tool Development.

To facilitate the human annotations of entity-landmark grounding, we develop a convenient web-based tool. Based on the label-studio platform, we design an annotator-friend interface targeted to our task, as presented in Figure 1. **Black** on the top line is a complete instruction, which consists of several sub-instructions. Firstly, the annotators can choose a pair of sub-instruction and sub-trajectory (sub-pair) to be marked sequentially. After a sub-pair is selected, the annotators should mark the entity words or phrases in the sub-instruction using different color labels, then mark the matched landmarks in the panoramas using the corresponding color bounding boxes. After marking all sub-pairs, the annotators submit the annotations of this episode and mark the next episode.



Turn left and exit out the door beside the TV to the left. Walk into the kitchen and go towards the right. Stop facing the 2 large double cabinets at the end of the kitchen to your right.

☑ 1[1] □ 2[2] □ 3[3] □ 4[4] □ 5[5] □ 6[6] □ 7[7]

Turn left and exit out the door beside the TV to the left.

Figure 1. The designed interface for entity-landmark grounding annotation. **Black** on the first line is the complete instruction. The several sub-pairs to be annotated are selected by "☐ #". The matched pair of the entity phrase and the landmark bounding box are marked with the same color label. The red arrow in the center of the panoramas denotes the next action direction.

A.3. Annotation Guideline Standardization.

In the data collection, five individuals with prior experience in visual grounding research served as our experts. After building the annotation tool, we adopt the pre-annotation to optimize our tool and standardize the annotation process. In the pre-annotation stage, our experts annotate 300 instruction-trajectory pairs together as examples and establish an annotation guideline and several rules based on their consensus after several discussions. Four rules are suggested to ensure the standardization of the annotation process.

- **Alignment Rule**: The entity phrase in the instruction should match the landmark panorama accurately.
- **Free Text Rule**: Free text instead of the class should be annotated, for instance, "the white dining table" instead of "table".
- **Text Coreference Rule**: The entity phrases referring

to the same object are marked with the same label.

- **Unique Landmark Rule**: For an entity phrase, only one corresponding landmark bounding box should be annotated in a panorama.

A.4. Data Annotation and Revision.

We first recruit 100 college students to annotate our dataset. Before starting our task, the students are asked to read the guideline and rules of the annotation carefully and attempt to annotate 50 instruction-trajectory pairs. Then we examine each annotated pair if the annotations highly agree with the four rules and reject participators with a low agreement. The qualification process leaves us with 43 qualified annotators to complete the annotation task. After an annotator finishes the annotations, our experts verify the annotations again and modify the inaccurate part to ensure that the annotations satisfy the four rules. In total, the annotation task costs more than 2000 hours and the revision task costs more than 1000 hours.

A.5. Data Processing.

To ensure annotation quality, we first reject the wrong annotations, i.e., alone entity annotations or landmark annotations, and then revise some wrong words in text annotations. Due to sub-pairs being annotated, the obtained positions of entity phrases are based on the sub-instructions. So we need to transfer the positions to the corresponding positions in the global instruction. On the other hand, we need to transfer the coordinates of the annotated bounding box to the corresponding coordinates in the panorama starting with 0 degrees. Finally, we combine the grounded entity-landmark annotations with the R2R dataset, obtaining the Grounded Entity-Landmark R2R (GEL-R2R) dataset.

B. Pre-trained Model

We adopt HAMT [3] as our per-trained model, which achieves the SoTA results in many VLN downstream benchmarks. Modified from the classical cross-modal model LXMERT [12], the HAMT inherits the fully transformer-based architecture. On the other hand, the HAMT designs a new hierarchical encoder to process history visual observations, which is considered important for decision-making in the long trajectory. Otherwise, to learn more effective initialization for VLN downstream tasks, the model is first pre-trained with several proxy tasks.

B.1. Model Architecture.

The architecture of the pre-trained model is illustrated in Sec. 4.2. The pre-trained model takes three inputs: a navigation instruction I, history information H_t, and current panoramic visual observation O_t. I is tokenized by using WordPieces first, and then feed into the language encoder, which is a multi-layer self-attention transformer following the standard BERT, to get a sequence word representation. H_t consists of all the past panoramic observations $\{o_{0,i}, \ldots, o_{t-1,i}\}_{i=1}^{36}$ and performed actions $\{act_0, \ldots, act_{t-1}\}$. This historic information is input into a history encoder, which has spatial encoding layers and temporal encoding layers. The spatiotemporal hierarchical encoder effectively represents history information as $\{h_{\text{cls}}, h_0, \ldots, h_{t-1}\}$, where h_{cls} is to learn a global hidden vector. O_t consists of image observations $v_{t,i}$ and orientation angle $a_{t,i}$. The pre-trained ViT [5] models encode $v_{t,i}$ as a 768-dimensional feature vector, which is concatenated with orientation embedding $(\sin\theta_{t,i}, \cos\theta_{t,i}, \sin\phi_{t,i}, \cos\phi_{t,i})$ to obtain the current visual state representation. And then cross-modal encoder, composed of self-attention layers and cross-attention layers, jointly encodes the features from the language and vision modality. Specifically, the visual modality is the concatenation of history and visual observation. As a result, the different modalities exchange the signals through cross-attention layers and align the token embedding with the same semantic information. Finally, the representations of tokens in instruction, history, and visual state are $Z = \{z_{\text{cls}}, z_1, \cdots, z_T\}$, $H_t = \{h_{\text{cls}}, h_1, \cdots, h_{t-1}\}$, $S_t = \{s_1, \cdots, s_{36}, s_{\text{stop}}\}$ respectively.

B.2. Pre-training Tasks.

As studied in previous work, transformer-based models in VLN are commonly pre-trained on the in-domain dataset using several proxy tasks to learn a more effective initialization representation for uni-modal and multi-modal information [7, 11, 3]. Common vision-language pre-training tasks and the VLN-specific auxiliary tasks are typically served as the proxy tasks. The HAMT model is pre-trained by five proxy tasks as follows.

Masked Language Modeling (MLM) [4]. MLM is a typical pre-training task for BERT-based models. In multi-modal transformer-based architecture, the task predicts masked words using surrounding words and image patches. It can facilitate the learned word representations to be grounded in the context of visual observations. Specifically, with a probability of 15%, we mask out the input words in the instruction I and replace them with a special token [MASK]. Based on their contextual textual and visual representations, the masked words are predicted via minimizing the negative log-likelihood of original words:

$$\mathcal{L}_{\text{MLM}} = -\log p\left(w_m \mid I_{\backslash m}, H_T\right), \tag{1}$$

where $I_{\backslash m}$ is the masked instruction, H_T is the complete trajectory.

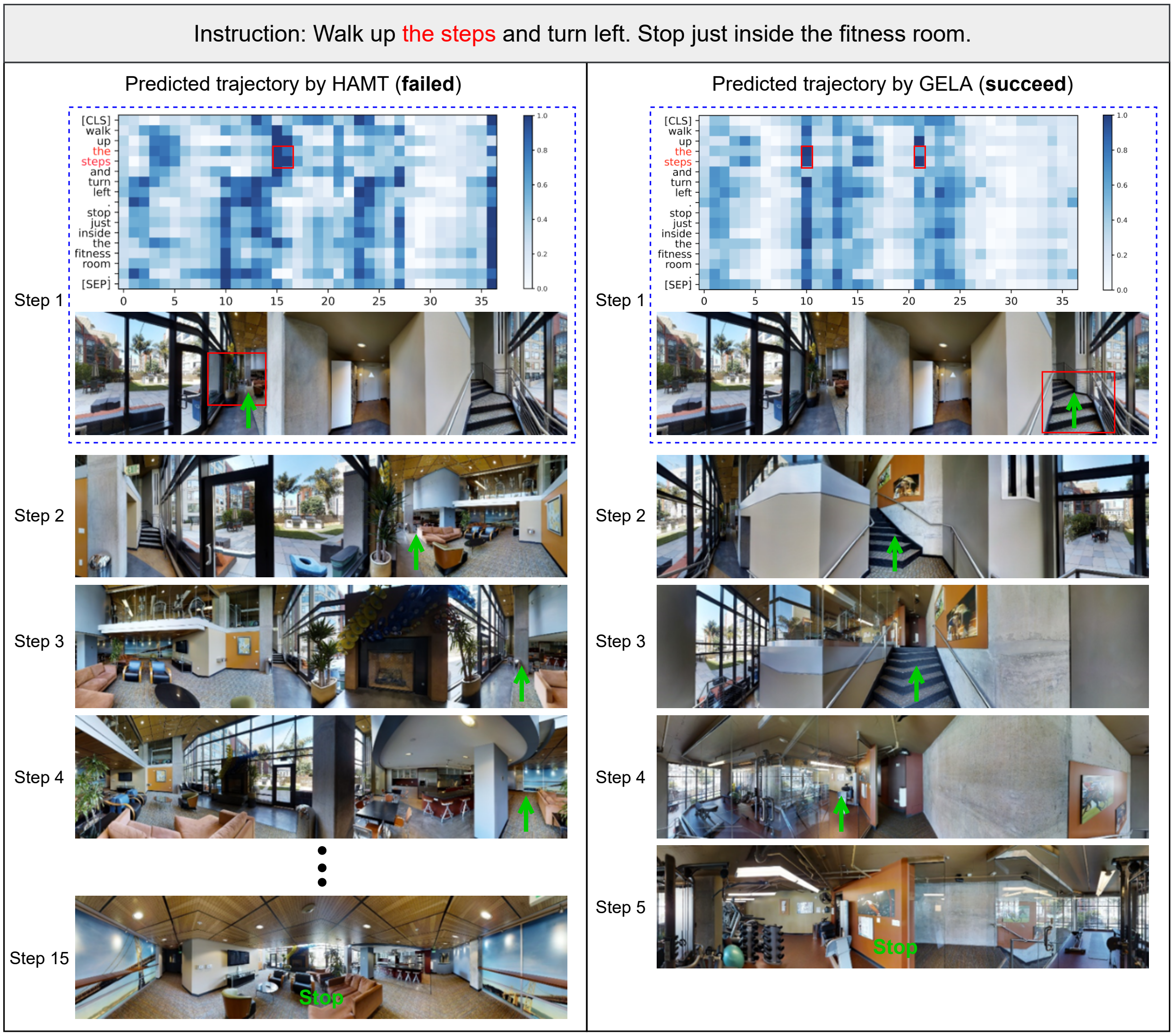

Instruction: Walk up the steps and turn left. Stop just inside the fitness room.
Predicted trajectory by HAMT (failed)
Predicted trajectory by GELA (succeed)
[CLS]
walk
up
the
steps
and
turn
left
.
stop
just
inside
the
fitness
room
[SEP]
0
5
10
15
20
25
30
35
1.0
0.8
0.6
0.4
0.2
0.0
Step 1
Step 2
Step 3
Step 4
Step 5
Step 15
Stop

Figure 2. Examples in R2R validation unseen split. The green arrow denotes the direction of the next action. Given the instruction on the top line, GELA and HAMT navigate in an environment. In the first step, GELA chooses the true direction but the HAMT chooses the wrong direction. The attention heatmaps at the last transformer layer in the cross-modal encoder are visualized above the panoramas of step 1. In GELA, "the steps" attend to the patches of the corresponding landmark (the red bounding box), but "the steps" in HAMT attend to other positions in the panorama. Therefore, recognizing "the steps" in step 1 helps GELA complete correct navigation.

Masked Region Classification (MRC) [9]. In analogy to MLM, MRC predicts the semantic class of masked image patches in the panorama based on instruction words and surrounding visual observations. It improves the ability of the model to understand the environments and match cross-modal information. Specifically, we zero out image patches in O_T with the probability of 15% as input. For the output embedding of masked patches, we predict the probability distribution P'_i on the 1000 classes of ImageNet. The objective is to minimize the KL divergence between P'_i and the supervisor P_i:

$$\mathcal{L}_{\text{MRC}} = -\sum_{j=1}^{1000} P_{i,j} \log P'_{i,j}, \tag{2}$$

where P_i is the predicted probability distribution by pre-trained ViT-B/16 [5].

Instruction Trajectory Matching (ITM) [10]. ITM is a particularly designed task for VLN, which distinguishes whether the input instruction-trajectory pairs match. It helps the model to learn the global cross-modal alignment

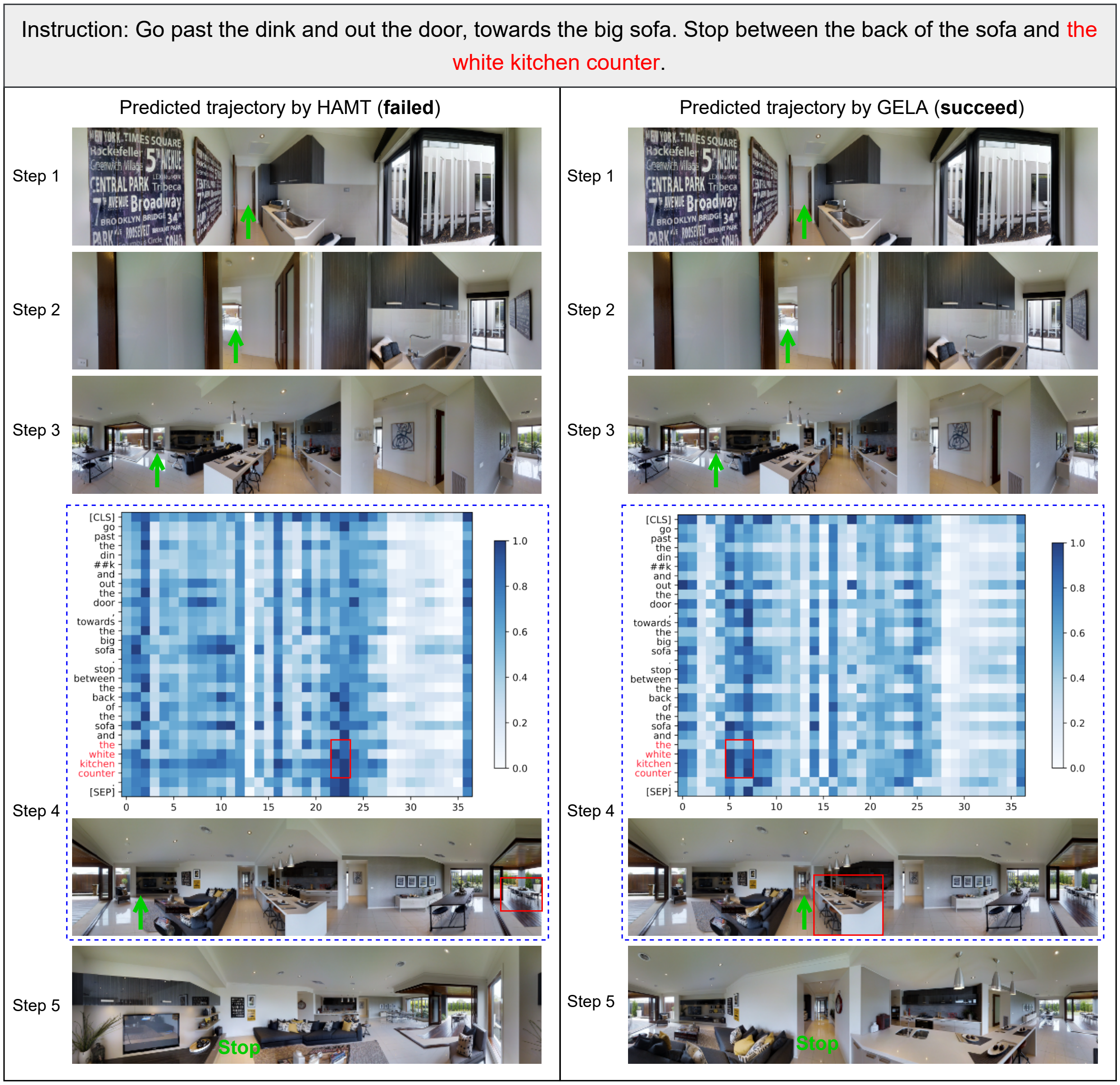


Figure 3. Examples in R2R validation unseen split. Given the instruction on the top line, GELA and HAMT navigate in an environment. GELA successfully reaches its destination. In the first three steps, HAMT chooses the right direction. However, in step 4, HAMT makes an error. The attention heatmaps at the last transformer layer in the cross-modal encoder are visualized above the panoramas of step 4. In GELA, "the white kitchen counter" attend to the patches of the corresponding landmark (the red bounding box). However, "the white kitchen counter" in HAMT attend to another similar landmark in the panorama, which results in the wrong action.

between the instructions and the overall temporal visual trajectory. Specifically, we sample four negative trajectories during pre-training for every positive instruction-trajectory pair. Two of the negative trajectories are chosen at random from other positive pairs in the mini-batch, and the other two are obtained by temporally rearranging the positive trajectory. We optimize this task via a Noisy Contrastive Estimation loss [6]:

$$\mathcal{L}_{\text{ITM}} = -\log \frac{\exp\left(g\left(I, H_T\right)\right)}{\exp\left(g\left(I, H_T\right)\right) + \sum_{k=1}^{4} \exp\left(g\left(I, H_{T,k}^{\text{neg}}\right)\right)}, \tag{3}$$

where $g\left(I, H_T\right)$ is the global matching score of I and H_T.

Single-step Action Prediction (SAP) [3]. SAP is a behavior cloning proxy task based on off-line expert demonstrations, which makes the learned representations benefit action decisions. The task predicts the next navigation action using instruction, history observations, and the current observation. Specifically, we apply a two-layer feedforward network (FFN) to predict action probability for each navigable view:

$$p_t\left(s_i'\right) = \frac{\exp\left(\mathrm{FFN}\left(s_i' \odot z_{\mathrm{cls}}'\right)\right)}{\sum_j \exp\left(\mathrm{FFN}\left(s_j' \odot z_{\mathrm{cls}}'\right)\right)}, \tag{4}$$

where $\odot$ is element-wise multiplication and z_{cls} is the output embedding of the special token [CLS]. We optimize this task by minimizing the negative log probability of the target visual state:

$$\mathcal{L}_{\mathrm{SAP}} = -\log p_t\left(s_{t+1}'\right). \tag{5}$$

Spatial Relationship Prediction (SPREL) [3]. SPREL is specially designed for spatial relations in navigation tasks. The task enhances the competence of the agent to identify directions by learning spatial relation aware representations. We predict the relative spatial position of two different views in a panorama only based on visual feature v_i, angle features a_i, or both $o_i = [v_i; a_i]$. Specifically, we randomly zero out v_i or a_i of the two views with a probability of 30%. The output embeddings of the two views are o_i' and o_j', and their relative heading and elevation angles are θ_{ij}, ϕ_{ij}. Then we predict $\theta_{ij}', \phi_{ij}' = FFN\left(\left[o_i'; o_j'\right]\right)$. We optimize this task via minimizing

$$\mathcal{L}_{\mathrm{SPREL}} = \left(\theta_{ij}' - \theta_{ij}\right)^2 + \left(\phi_{ij}' - \phi_{ij}\right)^2. \tag{6}$$

C. Qualitative Examples

Figure 2 and Figure 3 show trajectories predicted by our GELA model and compare them to results of the baseline model HAMT [3]. We see that GELA could better recognize the environment landmarks grounding the corresponding entity phrases in instructions.

References

[1] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian D. Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, pages 3674–3683, 2018. 1

[2] Angel X. Chang, Angela Dai, Thomas A. Funkhouser, Maciej Halber, Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from RGB-D data in indoor environments. In *3DV*, pages 667–676, 2017. 1

[3] Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. History aware multimodal transformer for vision-and-language navigation. In *NeurIPS*, pages 5834–5847, 2021. 1, 2, 4, 5

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186, 2019. 2

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2, 3

[6] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*, volume 9 of *JMLR Proceedings*, pages 297–304, 2010. 4

[7] 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 *CVPR*, pages 13134–13143, 2020. 2

[8] Yicong Hong, Cristian Rodriguez Opazo, Qi Wu, and Stephen Gould. Sub-instruction aware vision-and-language navigation. In *EMNLP*, pages 3360–3376, 2020. 1

[9] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, pages 13–23, 2019. 3

[10] 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 *ECCV*, volume 12351 of *Lecture Notes in Computer Science*, pages 259–274, 2020. 3

[11] 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. *CoRR*, abs/2203.11591, 2022. 2

[12] Hao Tan and Mohit Bansal. LXMERT: learning cross-modality encoder representations from transformers. In *EMNLP-IJCNLP*, pages 5099–5110, 2019. 2