Talk2Nav: Long-Range Vision-and-Language Navigation in Cities

Arun Balajee Vasudevan · Dengxin Dai · Luc Van Gool

Received: date / Accepted: date

Abstract Autonomous driving models often consider the goal as fixed at the start of the ride. Yet, in practice, passengers will still want to influence the route, e.g. to pick up something along the way. In order to keep such inputs intuitive, we provide automatic way finding in cities based on verbal navigational instructions and street-view images. Our first contribution is the creation of a large-scale dataset with verbal navigation instructions. To this end, we have developed an interactive visual navigation environment based on Google Street View; we further design an annotation method to highlight mined anchor landmarks and local directions between them in order to help annotators formulate typical, human references to those. The annotation task was crowdsourced on the AMT platform, to construct a new Talk2Nav dataset with 10,714 routes. Our second contribution is a new learning method. Inspired by spatial cognition research on the mental conceptualization of navigational instructions, we introduce a soft attention mechanism defined over the segmented language instructions to jointly extract two partial instructions – one for matching the next upcoming visual landmark and the other for matching the local directions to the next landmark. On the similar lines, we also introduce memory scheme to encode the local directional transitions. Our work takes advantage of the advance in two lines of research: mental formalization of verbal navigational instructions and training neural network agents for automatic way finding. Extensive experiments show that our method significantly outperforms previous navigation methods. For demo video, dataset and code, please refer to our project page.

Keywords Vision-and-language navigation · City navigation · Spatial Memory · Attention modeling

1 Introduction

Consider that you are traveling as a tourist in a new city and are looking for the famous coffee shop you would like to visit. You ask the locals and get a directional description "go ahead for about 200 meters until you hit a small intersection, then turn left and continue along the street before you see a yellow building on your right". People give indications that are not purely directional, let alone metric. They mix in referrals to landmarks that you will find along your route. This may seem like a trivial ability, as humans do this routinely. Yet, this is a complex cognitive task that relies on the development of an internal, spatial representation that includes visual landmarks (e.g. “the yellow building”) and possible, local directions (e.g. “going forward for about 200 meters”). Such representation can support a continuous self-localization as well as conveying a sense of direction towards the goal.

Just as a human can navigate in a new city when provided with navigational instructions, our aim is to teach an agent to perform the same task. If robots can find their way efficiently based on similar instructions, they will be able to reach their desired destination and perform their task with less effort for the humans giving them instructions. The robot then ‘understands’ the same levels of abstraction. The task is addressed recently as a Vision-and-Language Navigation (VLN) problem [4]. Although important progress was made, e.g. in constructing good datasets [3] and proposing effective learning methods [4, 20, 67, 48], this stream of work mainly focuses on synthetic worlds [30, 16, 12] or indoor room-to-room navigation [4, 20, 67]. Synthetic environments limit the complexity of the visual scenes while the room-to-room nav-
igation comes with the kind of challenges different from those of outdoors.

In order to learn long-range wayfinding in outdoor scenes, the first challenge lies with the creation of large-scale datasets. In order to be fully effective, the annotators providing the navigation instructions ought to know the environment like locals would. Training annotators to reach the same level of understanding for a large number of unknown environments is inefficient – in order to create one verbal navigation instruction, an annotator needs to search through hundreds of street-view images, remember their spatial arrangement, and summarize them into a sequence of route instructions. This straightforward annotation approach would be very time-consuming and error-prone. Because of this challenge, the state-of-the-art work uses synthetic directional instructions [31] or works mostly on indoor room-to-room navigation. For indoor room-to-room navigation, this challenge is less severe, due to two reasons: 1) the paths in indoor navigation are shorter; and 2) indoor environments have a higher density of ‘landmarks’. This makes self-localization, route remembering and descriptions easier. To our knowledge, there is only one other work from Chen et al. [13] on natural language based outdoor navigation, which also proposes an outdoor VLN dataset.

In order to address the data annotation challenge for our task, we develop an interactive visual navigation environment based on Google Street View, and more importantly design a novel annotation method which highlights selected landmarks and the spatial transitions in between. This enhanced annotation method makes it feasible to crowdsource this ‘intimidating’ annotation task. By hosting the tasks on the AMT platform, this work has constructed a new dataset Talk2Nav with 10,714 long-range routes within New York City (NYC).

The second challenge lies in training a wayfinding agent. Compared to indoor navigation, this learning task raises different challenges such as handling longer ranges, using cardinal directions, the placement, position and size of signs and visual landmarks, the position of the Sun, and the characteristics of traffic flows. Use in different environment also drastically changes the kind of language it elicits and requires fundamentally different reasoning. Inspired by the research on mental conceptualization of navigational instructions in spatial cognition [61,50,43], we introduce a soft attention mechanism defined over the segmented language instructions to jointly extract two partial instructions – one for matching the next coming visual landmark and the other for matching the spatial transition to the next landmark. Furthermore, the spatial transitions of the agent are encoded by an explicit memory framework which can be read from and written to as the agent navigates. One example of the outdoor VLN task can be found in Figure 1. Our work connects two lines of research that have been less explored together so far: mental formalization of verbal navigational instructions [61,50,43] and training neural network agent for automatic wayfinding [4,31].

Extensive experiments show that our method outperforms previous methods by a large margin. We also show the contributions of the sub-components of our method, accompanied with their detailed ablation studies. The collected dataset will be made publicly available upon the acceptance of the paper.

2 Related Works

Vision & Language Research at the intersection of language and vision has been conducted extensively in the last few years. The main topics include image captioning [38,72], visual question answering (VQA) [1,5], object referring expressions [7,9], grounded language learning [30,32] and among others. Although the goals are different from ours, some of the fundamental techniques are shared. For example, it is a common practice to represent visual data with CNNs pre-trained for image recognition and to represent textual data with word embeddings pre-trained on large text corpora. The main difference is that the perceptual input to the system is static while ours is active, i.e. the systems behavior changes the perceived input.

Vision Based Navigation Navigation based on vision and reinforcement learning (RL) has become a very interesting research topic recently. The technique has proven quite successful in simulated environments [54,78] and is being extended to more sophisticated real environments [53]. There has been active research on navigation-related tasks, such as localizing from only an image [69], finding the direction to the closest McDonalds, using Google Street View Images [40,11], goal based visual navigation [26] and others. Gupta et al. [26] uses a differentiable mapper which writes into a latent spatial memory corresponding to an egocentric map of the environment and a differentiable planner which uses this memory and the given goal to give navigational actions to navigate in novel environments. There are few other recent works on vision based navigation [58,70]. Thoma et al. [58] formulates compact map construction and accurate self localization for image based navigation by careful selection of suitable visual landmarks. Recently, Wortsman et al. [70] proposes a meta-reinforcement learning approach for visual navigation, where the agent learns to adapt in unseen environments in a self-supervised manner.

Vision-and-Language Navigation Here, the task is to navigate an agent in an environment to a particular destination based on language instructions. The following are some recent works in Vision-and-Language Navigation (VLN) [4,67,20,66,55,48,39] task. The general goal of these works
Fig. 1: An illustration that an agent finding its way from a source point to a destination. The left shows the segmented navigational instructions with ‘red’ indicating visual landmark descriptions and ‘blue’ the local directional instructions. The right panel shows the agent’s local observations, the memory of traversed route and its decision of moving forward.

are similar to ours – to navigate from a starting point to a destination in a visual environment with language directional descriptions. Anderson et al. [4] created the R2R dataset for indoor room-to-room navigation and proposed a learning method based sequence-to-sequence neural networks. Subsequent methods [67, 66] applies reinforcement learning and cross-modal matching techniques on the same dataset. The same task was tackled by Fried et al. [20] using speaker-follower technique to generate synthetic instructions for data augmentation and pragmatic inference. While sharing similarity, our work differs significantly from them. The environment domain is different; as discussed in Section 1, long-range navigation in cities raises very different challenges than indoor navigation both in data annotation and training the agent. There are concurrent works aiming at extending Vision-and-Language Navigation to city environment [31, 13, 41]. The difference to Hermann et al. [31] lies in that our method works with real navigational instructions, instead of the synthetic ones summarized by Google Maps. This difference leads to different tasks and in turn to different solutions. Kim et al. [41] proposes end-to-end driving model that takes natural language advice to predict control commands to navigate in city environment. Chen et al. [13] proposes outdoor VLN dataset similar to ours, where real instructions are created from Google Street View\(^1\) images. However, we differ in the way we decompose our navigational instructions to make our dataset annotation easier. Our annotation method draws inspiration from spatial cognition field to specifically promote annotators’ memory and thinking, making the task less energy-consuming and less error-prone. We shall see more details in Section 3.

Visual landmarks for Navigation. There are numerous studies in cognition and psychology which state the significance of using visual landmarks in route descriptions [33, 37, 60]. They show that route descriptions consist of descriptions for visual landmarks and local directional instructions between consecutive landmarks [52, 50]. Similar techniques – a combination of visual landmarks, as rendered icons, and highlighted routes between consecutive landmarks – are constantly used for making efficient maps [61, 68, 22]. It has also been shown that topological view of the environment helps in translating natural language to actions for navigation behaviors [75].

Attention & Memory for Language Modeling Attention mechanism has been used widely for language [49] and visual inputs [73, 65, 3]. Language attention mechanism has been shown to produce state-of-the-art results in machine translation [8] and other natural language processing tasks like VQA [36, 74, 34], image captioning [6], grounding reference expressions [34, 35] and others. MAC from Hudson et al. [36] has a control unit which performs soft attention-based weighted average of the question words and other units that performs multiple read and write operations and later extract information from images for VQA task. Hu et al. [34] also has similar language attention mechanism but they decompose the reasoning into sub-tasks/modules and predict modular weights from input text. Attention mechanism is one of the main component for the top-performing algorithms such as Transformer [62] and BERT [18] in NLP tasks. In our model, we adopt soft attending over linear memory features from the work of Kumar et al. [44] while we apply the soft attention over segmented language instructions to attend over a pair of sub-instructions: a) for landmark and b) for local directions.

There are generally two kinds of memory used in the literature: a) implicit memory and b) explicit memory. Implicit memory learns to memorize knowledge in the hidden state vectors via back-propagation of errors. Typical examples include RNNs [38] and LSTMs [19]. Explicit memory, however, features explicit read and write modules with attention mechanism. Notable examples are Neural Turing Machines [24] and Differentiable Neural Computers (DNCs) [25].
In our work, we use external explicit memory in the form of a memory image which is accessed by its read and write modules. Training a soft attention mechanism over language segments coupled with an explicit memory scheme makes our method more suitable for long-range navigation where the reward signals are sparse.

3 Talk2Nav Dataset

The target is to navigate using language descriptions in real outdoor environment. Recently, numerous datasets on language based visual navigation task have been released both on indoor [4] and outdoor [10,13] environments. Existing datasets typically have one overall language description for the entire path/route, lacking the correspondence between language descriptions and sub-units of a route. This poses challenges in learning long-range vision-and-language navigation (VLN). To address this issue, this work proposes a new annotation method and uses it to create a new dataset Talk2Nav.

Talk2Nav contains navigation routes at city levels. A navigational city graph is created with nodes as locations in the city and connecting edges as the roads between the nodes, similar to Mirowski et al. [53]. Each route as shown in Figure 2 from a source node to a destination node is composed of densely sampled atomic unit nodes, each containing a) street-view panoramic image, b) GPS coordinates, c) bearing angles. Furthermore, we enrich the routes with intermediary visual landmarks, language descriptions for these visual landmarks and the local directional instructions connecting the landmarks.

3.1 Data Collection

To retrieve city route data, OpenStreetMap is used for getting metadata information of locations and Google’s APIs are used to obtain maps and street-view images.

Path Generation. The path from a source to a destination is sampled from a city graph. To that aim, we used OpenStreetMap which provides latitudes, longitudes and bearing angles of all the locations (waypoints) within a predefined region in the map. A city graph is defined by taking the locations of the atomic units as the nodes, and the directional connections between neighbourhood locations as the edges. The K-means clustering algorithm (k=5) is applied to the spatial locations (GPS coordinates) of all nodes. We then randomly picked up two nodes from different clusters to ensure that the source node and the destination node are not too close. The A* search algorithm is then used generate a path by finding the shortest traversal path from the source to destination in the city graph.

Fig. 2: An illustrative route from a source node to a destination node with 4 landmarks labelled along the route. Landmark 4 is same as the destination node. Sub-routes are defined as routes between the consecutive landmarks.

Street View Images. We collect the 360° street-view images along with its heading angle (yaw_deg) with the help of Google Street View API and the metadata. The API allows for downloading tiles of street-view panoramic images which are then stitched together as an equirectangular projection image. We use the heading angle to re-render the street-view panorama images such that the images are centre-aligned to the heading direction of the route.

3.2 Directional Instruction Annotation

The main challenge of data annotation for automatic language-based wayfinding lies in the fact that the annotators need to play the role of an instructor as the local people do to tourists. This is especially challenging when the annotators do not know the environment well. The number of street-view images for a new environment is tremendous and searching through them can be costly, not to mention remembering and summarizing them to verbal directional instructions.

Inspired by the large body of work in cognitive science on how people mentally conceptualize route information and convey routes [42,50,61], our annotation method is designed to specifically promote memory or thinking of the annotators. For the route directions, people usually refer to visual landmarks [33,50,59] along with a local directional instruction [61,63]. Visualizing the route with highlighted salient landmarks and local directional transitions compensates for limited familiarity or understanding of the environment.

3.2.1 Landmark Mining

The choice of landmarks along the route is a subjective task. We frame the task as a summarization problem using submodular optimization to create summaries that takes into account multiple objectives. In this work, three criteria are considered: 1) the selected images are encouraged to spread

2 http://maps.google.com/cbk?output=xml&ll=40.735357,-73.918551&dm=1
out along the route to support continuous localization and guidance; 2) images close to road intersections and the approaching side of the intersections are preferred for better guidance through intersections; and 3) images which are easy to be described, remembered and identified are preferred for effective communication.

Given the set of all images $I$ along a path $P$, the problem is formulated as a subset selection problem that maximizes a linear combination of the three submodular objectives:

$$L = \arg \max_{L' \subseteq \varphi(I)} \sum_{i=1}^{3} w_i f_i(L', P), \quad \text{s.t.} \quad |L'| = l \tag{1}$$

where $\varphi(I)$ is the powerset of $I$, $L'$ is the set of all possible solutions for the size of $l$, $w_i$ are non-negative weights, and $f_i$ are the sub-modular objective functions. More specifically, $f_1$ is the minimum travel distance between any of the two successive selected images along the route $P$; $f_2 = 1/(d + \sigma)$ with $d$ the distance to the closest approaching intersection and $\sigma$ is set to 15 meters to avoid having an infinitely large value for intersection nodes; $f_3$ is a learned ranking function which signals the easiness of describing and remembering the selected images. The weights $w_i$ in Equation 1 are set empirically: $w_1 = 1$, $w_2 = 1$ and $w_3 = 3$. $l$ is set to 3 in this work as we have source and destination node to be fixed and we choose three landmarks in between as shown in Figure 2. The model $f_3$ is presented next.

**Ranking model.** In order to train the ranking model for images of being ‘visual landmarks’ that are easier to describe and remember, we compile images from three cities: New York City (NYC), San Francisco (SFO) and London covering different landscapes such as high buildings, open fields, downtown areas, etc. We select 20,000 pairs of images from the compiled set. A pairwise comparison is performed over 20,000 pairs to choose one over the other. We crowd-sourced the annotation with the following criteria: a) Describability – how easy to describe it by the annotator, b) Memorability – how easy to remember it by the agent (e.g. a traveler such as a tourist), and c) Recognizability – how easy to recognize it by the agent. We learn the ranking model with a Siamese network [14] by following [27]. The model takes a pair of images and scores the selected image more than the other one. We use the Huber rank loss as the cost function. In the inference stage, the model ($f_3$ in Equation 1) outputs the averaged score for all selected images signalling their suitability as visual landmarks for the VLN task.
3.2.2 Annotation and Dataset Statistics

In the literature, a few datasets have been created for similar tasks [4, 13, 64, 75]. For instance, Anderson et al. [4] annotates the language description for the route by asking the user to navigate the entire path in egocentric perspective. Incorporation of overhead map of navigated route as an aid for describing the route can be seen in [13, 64, 75]. In our route description annotation process, the mined visual landmarks are provided, along with an overhead topological map and the street-view images on the left. We crowd-source this annotation task on Amazon Mechanical Turk. The annotator is shown a pre-defined route on the GoogleMap as shown in Figure 3. The green line denotes the complete route while red line shows the current road segment. We can use Move forward and Move backward button to navigate from the source node to the destination node. The annotator is instructed to watch the 360° Street View images on the left. Here, we have customized Google Street View interface to allow the annotator to navigate along the street-view images simultaneously as they move forward/backward in the overhead map. The street view is aligned to the direction of navigation such that forward is always the moving direction. To minimize the effort of viewing Street View, we also provide four perspective projected images: left-view, front-view, right-view, and rare-view.

The annotator can navigate the complete route comprising of \( m \) landmark nodes and \( m \) intermediate sub-paths and is asked to provide descriptions for all landmarks and for the local directional navigation of all sub-paths. In this work, we use \( m = 4 \), which means that we collect 4 landmark descriptions and 4 local directional descriptions for each route as shown in Figure 2. Shorter routes of varying lengths can be sampled from the collected data.

**Statistics.** We gathered 43,630 locations which include GPS coordinates (latitudes and longitudes), bearing angles, etc. in New York City (NYC) covering an area of 10km x 10km as shown in Figure 4. Out of all those locations, we managed to compile 21,233 street-view images (outdoor) – each for one road node. We constructed a city graph with the locations as nodes with around 80,000 edges. We annotated 10,714 navigational routes and the corresponding route descriptions, as detailed in Section 3.1. These route descriptions are composed of 34,930 node descriptions and 27,944 local directional descriptions. The average length of the navigational instructions, the landmark descriptions and the local directional instructions of the sub-paths are 70 words, 8 words and 7 words, respectively. In contrast, the average length of the language descriptions of the Room-to-Room (R2R) dataset is 29 words [4], which is notably shorter than ours. In total, our dataset Talk2Nav contains 5,240 unique words having four or more occurrences compared to 3,156 in the
Table 1: Comparison of Talk2Nav dataset with R2R dataset [4] under different measures.

| Criteria                               | R2R [4] | Talk2Nav |
|----------------------------------------|---------|----------|
| # of navigational routes               | 7,189   | 10,714   |
| # of panoramic views                   | 10,800  | 21,233   |
| # of navigational descriptions        | 21,567  | 10,714   |
| # of navigational descriptions per path| 3       | 1        |
| Average length of navigational descrip-| 29      | 70       |
| tions                                 |         |          |
| # of landmark descriptions            | -       | 34,930   |
| Average length of landmark descrip-    | -       | 8        |
| tions                                 |         |          |
| # of local directional descrip-        | -       | 27,944   |
| tions                                 |         |          |
| Average length of local directional   | -       | 7.2      |
| descrip-                              |         |          |
| tions                                 |         |          |
| Vocabulary Size                       | 3,156   | 5,240    |

R2R dataset. Figure 5 shows the distribution of the length (number of words) of the landmark descriptions, the local directional instructions and the complete navigational instructions. In Table 1, we show a more detailed comparison with the R2R dataset [4] under different criteria. In terms of payment, we paid $0.01 and $0.5 for the landmark ranking task and the route description task, respectively. We have a qualification test. Only the annotators who passed the qualification test were employed for the real annotation. In total, 150 qualified workers were employed.

4 Approach

We create a street-view environment on top of the compiled city graph of Talk2Nav dataset. The city graph consists of 21,233 nodes with the corresponding street-view images (outdoor) and 80,000 edges representing the travelable road segments between the nodes. During training and testing stages, we use this simulator to navigate the agent in the Street View environment using the city graph. Based on the predicted action at each node, the agent moves to the next node in the environment.

Our system is a single agent traveling in an environment represented as a directed connected graph. Each node contains a 360° panoramic image, a location (defined by latitudes and longitudes) and bearing angles of the connecting roads. A valid route is a sequence of connected edges from a source node to a destination node. A successful navigation is defined as when the agent correctly follows the right route referred by the natural language instruction. In short, the environment has discrete set of state spaces as nodes in the city graph. The prospective future states of the agent of the environment are a function of the actions made by the agent. For the sake of tractability, we assume no uncertainty (such as dynamic changes in the environment, noise in the actuators) in the environment, hence it is a deterministic goal setting. 4.1 Route Finding Task

The task defines the goal of an embodied agent to navigate from a source node to a destination node based on a navigational instruction. Specifically, given a natural language instruction \( X = \{x_1, x_2, ..., x_n\} \), the agent needs to perform a sequence of actions \( \{a_1, a_2, ..., a_m\} \) from action space \( \mathcal{A} \) to hop over nodes in the environment space to reach the destination node. When the agent executes an action, it interacts with the environment and receives a new visual observation. The agent performs this kind of action sequentially until it reaches the destination node successfully or fails the task because it exceeds the maximum episode length.

The agent learns to predict an action at each state to navigate in the environment. Learning long-range vision-language navigation (VLN) requires an accurate sequential matching between the navigation instruction and the route. As argued in the introduction, we observe that navigation instructions consist of two major classes: landmark descriptions and local directional instructions between landmarks. In this work, given a language instruction, our method segments it to a sequence of two interleaving classes: landmark descriptions and local directional instruction. Please see Figure 1 for an example of a segmented navigational instruction. As it moves, the agent learns an associated reference position in the language instruction to obtain a softly-attended local directional instruction and landmark description. When one landmark is achieved, the agent updates its attention and moves towards the new goal, i.e. the next landmark. Two matching modules (MM as shown in Figure 6) are used to score each state: 1) between the traversed path in memory and the local, softly-attended directional instruction, and 2) between the visual scene the agent observes at the current node and the local, softly-attended landmark description. An explicit external memory is used to store the traversed path from the latest visited landmark to the current position.

Each time a visual observation is successfully matched to a landmark, a learned controller increments the reference position of the soft attention map over the language instruction to update both the landmark and directional descriptions. This process continues until the agent reaches the destination node or the reference position on the instruction exceeds the length or the episodic length of the navigation is exceeded. A schematic diagram of the method is shown in Figure 6. Below we detail all the models used by our method.

4.1.1 Language

We segment the given instruction to two classes: a) visual landmark descriptions and b) local directional instructions between the landmarks. We employ the BERT transformer
model [18] to classify a given language instruction $X$ into a sequence of token classes $\{c_i\}_{i=1}^n$ where $c_i \in \{0,1\}$, with 0 denoting the landmark descriptions and 1 denoting the local directional instructions. By grouping consecutive segments of the same token class, the whole instruction is segmented into interleaving segments. An example of the segmentation is shown in Figure 1 and multiple more in Figure 8. Those segments are used as the basic units of our attention scheme rather than the individual words used by previous methods [4]. This segmentation converts the language description into a more structured representation, aiming to facilitate the language-vision and language-trajectory matching problem. Denoted by $T(X)$ a sequence of segments of landmark and local directional instructions, then

$$T(X) = \{(L^1, D^1), (L^2, D^2), \ldots, (L^J, D^J)\}$$

(2)

where $L^j$ denotes the feature representation for landmark description segment $j$, and $J$ is the total number of segments in the route description.

As it moves in the environment, the agent is associated with a reference position in the language description in order to put the focus on the most relevant landmark descriptions and the most relevant local directional instructions. This is modeled by a differentiable soft attention map. Let us denote by $\eta_t$ the reference position for time step $t$. The relevant landmark description at time step $t$ is extracted as

$$\bar{L}^\eta_t = \sum_{j=1}^{J} L^j e^{-|\eta_t-j|}$$

(3)

and the relevant directional instruction as

$$\bar{D}^\eta_t = \sum_{j=1}^{J} D^j e^{-|\eta_t-j|}$$

(4)
As the agent navigates in the environment, we have a write module that traces the path from topological view using the sequence of GPS coordinates of the traversed path. Our write module traces the path travelled from the latest visited landmark to the current position. The path is rasterized into an image. As to the rasterization, red lines are used to represent the path, blue square marks are used to denote the source, and blue disk markers are used for the current location of the agent. The write module always writes from the centre of the memory image. Whenever the coordinates of the new rasterized pixel are beyond the image dimensions, the module increases the scale of the map until the new pixel is in the image and has a distance of 10 pixels to the boundary. An image of $200 \times 200$ pixels is used and the initial scale of the map is set to 5 meters per pixel. Please find examples of the memory images in Figure 7.

Each memory image is associated with the value of its scale (metres per pixel). Deep features $\psi_2(M_t)$ are extracted from the memory image $M_t$ which are then concatenated with its scale value. The concatenated features are passed to the matching module. The matching module verifies the semantic similarity between the traversed path and the provided local directional instruction. The concatenated features are also provided to the action module along with the local directional instruction features to predict the action for the next step.

### 4.1.4 Matching Module

Our matching module is used to determine whether the aimed landmark is reached. As shown in Figure 6, the matching score is determined by two complementary matching modules: 1) between the visual scene $\psi_1(I_t)$ and the extracted landmark description $L^m_t$ and 2) between the spatial memory $\psi_2(M_t)$ and the extracted directional instruction $D^m_t$.

For both cases, we use a generative image captioning model and compute the probability of reconstructing the language description given the image. Scores are the averaged generative probability over all words in the instruction. Let $s^1_t$ be the score for pair $(\psi_1(I_t), L^m_t)$ and $s^2_t$ be the score for pair $(\psi_2(M_t), D^m_t)$. We then compute the score feature $s_t$ by concatenating the two:

$$\mathbf{s}_t = (s^1_t, s^2_t).$$

The score feature $s_t$ is fed into a controller $\phi(.)$ to decide whether the aimed landmark is reached:

$$\phi_t(s_t, h_{t-1}) \in \{0, 1\},$$

where 1 indicates that the aimed landmark is reached and 0 otherwise, $\phi_t(.)$ is an Adaptive Computation Time (ACT) LSTM [23] which allows the controller to learn to make decisions at variable time steps, $h_{t-1}$ is the hidden state of

$$\eta_{t+1} = \eta_t + \phi_t(.) \text{ and } \eta_0 = 1.$$ (5)

$\phi_t(.)$ is an Indicator Controller learned to output 1 when a landmark is reached and 0 otherwise. The controller is shown in Figure 6 and is defined in Section 4.1.4. After a landmark is reached, $\eta_t$ increments 1 and the attention map then centers around the next pair of landmark and directional instructions. We initialize $\eta_0$ as 1 to position the language attention around the first pair of landmark and directional instruction.

### 4.1.2 Visual Observation

The agent perceives the environment with an equipped 360° camera which obtains the visual observation $I_t$ of the environment at time step $t$. From the image, a feature $\psi_1(I_t)$ is extracted and passed to the matching module as shown in Figure 6 which estimates the similarity (matching score) between the visual scene $I_t$ and a softly attended landmark description $\bar{L}^n_t$ which is modulated by our attention module. The visual feature is also passed to the action module to predict the action for the next step as shown in Section 4.1.5.

### 4.1.3 Spatial Memory

Inspired by [21,25], an external memory $M_t$ explicitly memorizes the agent’s traversed path from the latest visited landmark. When the agent reaches a landmark, the memory $M_t$ is reinitialized to memorize the traversed path from the latest visited landmark. This reinitialization can be understood as a type of attention to focus on the recently traversed path in order to better localize and to better match against the relevant directional instructions $D^m_t$ modeled by the learned language attention module defined in Section 4.1.1.
the controller. In this work, \( \phi_t(\cdot) \) learns to identify the landmarks with the variable number of intermediate navigation steps.

### 4.1.5 Action Module

The action module takes the following inputs to decide the moving action: a) the pair of \((\psi_1(L_i), L^m)\), b) the pair of \((\psi_2(M_t), D^m)\), and c) the matching score feature \(s_t\) as defined in Equation 6. The illustrative diagram is shown in Figure 6.

The inputs in a) above is used to predict \(a^e_t\) – a probability vector of actions over the action space. The inputs in b) is used to predict \(a^m_t\) – the second probability vector of actions over the action space. The two probability vectors are combined by a weighted average, with weights learned from the score feature \(s_t\). Specifically, \(s_t\) is fed to a FC network to output the weights \(w \in \mathbb{R}^2\). For both a) and b), we use encoder LSTM as in [4] to encode the language instruction \(D^m\). We then concatenate the encoder’s hidden states with the image encodings (i.e. \(\psi_1(L_i)\) and \(\psi_2(M_t)\)) and pass through a FC network to predict the probability distribution \(a^e_t\) and \(a^m_t\). We make final action prediction \(a_t\) as:

\[
a_t = \frac{1}{\sum_i w_i} (w_0 * a^e_t + w_1 * a^m_t).
\]

The action space \(A\) is defined as follows. We divide the action space \(A\) into 8 directions. Each direction is centred at \((i \times 45^\circ) : i \in [0, ..., 7]\) with \(\pm 22.5^\circ\) offset as illustrated in Figure 6. When an action angle is predicted, the agent turns to the particular angle and moves forward to the next node along the road. The turning and moving-forward define an atomic action. The agent comes to a stop when it encounters the final landmark.

### 4.2 Learning

The model is shown in Figure 6. It is trained in a supervised way. We followed the student-forcing approach proposed in [4] to train our model. At each step, the action module is trained with a supervisory signal of the action in the direction of the next landmark. This is in contrast with previous methods [4], in which it is the direction to the final destination.

We use cross entropy loss to train the action module and the matching module as they are formulated as classification tasks for action prediction and conditioned generation of navigation instructions, respectively. For ACT model, we use weighted binary cross entropy loss at every step (higher weight for positives). The supervision of the positive label (‘1’) for ACT model comes into effect only if the agent reaches the landmark which is sporadic. In short, our total loss is a summation of the action module loss functions with equal weights:

\[
Loss = Loss_{action} + Loss_{landmark matching} + Loss_{direction matching} + Loss_{ACT}.
\]

The losses of the two matching modules take effect only at the place of landmarks which are much sparser than the road nodes where the action loss and ACT loss are computed. Because of this, we first train the matching networks individually for the matching tasks, and then integrate them with other components for the overall training.

### 4.3 Implementation Details

#### Language Module

We use the BERT [18] transformer model pretrained on BooksCorpus [77] and English Wikipedia\(^3\), for modelling language. This yields contextual word representations which is different from classical models such as word2vec [51], GloVe [56] which are context-free. We use a word-piece tokenizer to tokenize the sentence, by following [18, 71]. Out of vocabulary words are split into sub-words based on the available vocabulary words. For word token classification, we first train BERT transformer with a token classification head. Here, we use the alignment between the given language instruction \(X\) and their corresponding

---

\(^3\) https://en.wikipedia.org/wiki/English_Wikipedia
set of landmark and directional instruction segments in the train split of Talk2Nav dataset. We then train the transformer model to classify each word token in the navigational instruction to be a landmark description or a local directional instruction.

At the inference stage, the model predicts a binary label for each word token. We later convert the sequence of word tokens into segments \( T(X) \) by simply grouping adjacent tokens which have the same class. We note that this model has a classification accuracy of 91.4% on the test set. We have shown few word token segmentation results in Figure 8.

**Visual inputs.** The Google street-view images are acquired in the form of equirectangular projection. We experimented with SphereNet [15] architecture pretrained on the MNIST [45] dataset and ResNet-101 [28] pretrained on ImageNet [17] to extract \( \psi_1(I) \). Since MNIST pretrained SphereNet is relatively a shallow network, we adapted SphereNet architecture to suit StreetView images by adding more convolutional blocks. Later, we define a pretext task using the street-view images from Talk2Nav dataset to learn SphereNet weights. Given two street view images with an overlap of their visible view, the task is to predict the difference between their bearing angles and the projection of line joining the locations on the bearing angle of second location. We frame the problem as a regression task of predicting the above two angles. This encourages SphereNet to learn the semantics in the scene. We compiled the training set for this pre-training task from our own training split of the dataset. In the case of memory image, we use ResNet-101 pretrained on ImageNet to extract \( \psi_2(M) \) from the memory image \( M \).

**Other modules.** We perform image captioning in the matching modules using the transformer model, by following [46, 76]. We pretrain the transformer model for captioning in the matching module using the landmark street-view images and their corresponding descriptions from the training split of Talk2Nav for matching of the landmarks. For the other matching module of local directions, we pretrain the transformer model using the ground truth memory images and their corresponding directional instructions. We synthesized the ground truth memory image in the same way as our write module in agent’s external memory (as mentioned in Section 4.1.3). We finetune both the match modules in the training stage. All the other models such as Indicator Controller, Action module are trained in an end-to-end fashion.

**Network details.** The SphereNet model consists of five blocks of convolution and max-pooling layers, followed by a fully-connected layer. We use 32, 64, 128, 256, 128 filters in the 1 to 5 convolutional layers and each layer is followed by a Max pooling and ReLU activation. This is the backbone structure of SphereNet. For ResNet, we use ResNet-101 as the backbone. Later, we have two branches: the fully connected layer has 16 neurons using a softmax activation function in one branch of the network for the action module and 512 neurons with ReLU activation in the other branch for the match module. Hence, input of image feature size from these models are of 512 and attention feature size over the image is 512. The convolutional filter kernels are of size 5x5 and are applied with stride 1. Max pooling is performed with kernels of size 3x3 and a stride of 2. For language, we use an input encoding size of 512 for each token in the vocabulary. We use Adam optimizer with learning rate of 0.01 and alpha and beta for Adam as 0.999 and \( 10^{-8} \). We trained 27, 944 landmark descriptions with their street-view images for 20 epochs with mini-batchsize of 16.

5 Experiments

Our experiments focus on 1) the overall performance of our method when compared to the state-of-the-art (s-o-t-a) navigation algorithms, and 2) multiple ablation studies to further understand our method. The ablation studies cover a) the importance of using explicit external memory, b) performance evaluation of navigation at different levels of difficulty, and c) a comparison of visual features. We train and evaluate our model on the Talk2Nav dataset. We use 80% of the dataset for training and the rest for testing. There is no overlap between the training and the testing environments. By following [2], we evaluate our method under three metrics:

- **SPL**: Success Rate weighted by Normalized Inverse Path Length. It penalizes the successes made with longer paths.
- **Navigation Error**: The distance to the goal after finishing the episode.
- **Average Steps**: The average number of steps required to reach the goal successfully.

We compare with the following s-o-t-a navigation methods: Student Forcing [4], RPA [67], Speaker-Follower [20] and Self-Monitoring [47]. We trained their models on the same data that our model uses. In addition, we add one more comparison to Oracle where we use the ground truth landmark and local directional instructions as input to the match module. We also study the performance of our complete model and other s-o-t-a methods at varying difficulty levels: from short navigation paths consisting of one landmark to long ones consisting of four landmarks.

In order to evaluate methods at different difficulty levels, we generate datasets of different navigation difficulties from the Talk2Nav dataset. The navigation difficulty of a route is approximated by the length of the route which is measured by the number of landmarks it contains. In particular, in Talk2Nav, each route consists of 4 landmarks. For our primary experiments, we use the whole route for training and testing. For specific experiments to evaluate at different
difficulty levels, we sub-sampled routes with the length of 1, 2 and 3 landmarks from the annotated 4-landmark routes. For instance, we sample four 1-landmark sub-routes, three 2-landmark sub-routes and two 3-landmark sub-routes from a 4-landmark route. We also generate a dataset with all four levels of navigation difficulty by mixing the original Talk2Nav and the three generated datasets. We use these sub-sampled routes to generate the datasets for the cross-difficulty evaluation. Let us focus on the experiments one by one.

5.1 Comparison to Prior works

To make a fair comparison with prior works, we use the same image features and language features in all the cases. We use pre-trained ResNet-101 model on ImageNet to extract image features and pre-trained BERT transformer model for language features. For Self-Monitoring [47] and Speaker-Follower [20], panoramic view of the environment is discretized into 8 view-angles (8 headings x 1 elevation with 45 degree intervals). The navigable directions at each location are defined by the city graph of Talk2Nav dataset. We use greedy action selection during evaluation as beam search decoding for action selection leads to lower SPL because of longer trajectory lengths [48].

Table 2 shows the results of our method and other competing s-o-t-a methods. We tabulate results under all the evaluation metrics. The destination threshold (when the distance of the final reached position and the destination node is within this range, it is a successful navigation) is set to 100m and the trajectory length (denoting additional allowed path length w.r.t ground truth path length to destination) is set to 30%. The row for Oracle in Table 2 denotes the maximum accuracy that could be achieved when the ground-truth segments of the instructions are used instead of segments by the trained segmentation method in Section 4.1.1.

The table shows that our method achieves significantly better results than other existing methods under all considered evaluation metrics for long-range Vision-and-Language navigation in outdoor environments. For instance, our method improves SPL from 9.56% to 11.92%, reduces Navigation Error from 740.12m to 633.85m and reduces Average Steps from 13.01 to 12.51, when compared to the previous best performing method [47].

In addition to the evaluation for the long-range (i.e. 4-landmark) navigation, we also study the performance of all these trained methods when evaluated at varying difficulty levels: from short navigation paths consisting of one landmark to long ones consisting of four landmarks. We evaluate under SPL and compare our method with the prior works as before. The results are listed in Table 3. We observe that our method outperforms all the other prior works by a large margin. For instance, our method improves at SPL (%) from 72.21 to 74.28 for short routes having 1 landmark, from 37.71 to 43.08 for routes with 2 landmarks, and from 23.16 to 27.96 for routes with 3 landmarks, when compared to [47].

The reason behind the good performance of our method can be attributed to multiple factors. The decomposition of the whole navigation instruction into landmark descriptions and local directional instructions, the attention map defined on language segments instead of English words, and the two clearly purposed matching modules make our method suitable for long-range vision-and-language navigation. Due to these introduced components and the design that allows them to work together, the agent is able to put the focus to the right place and does not get lost easily as it moves.

Previous methods aim to find a visual image match for the given language sentence at each step to predict an action. We argue that this is not optimal. The navigational instructions indeed consist of mixed referrals for visual content (landmarks) and for spatial movements. For instance, it is wrong to match a sentence like ‘Go forward 100m and take a left turn’ to an image observation. Our method distinguishes the two types of language sentences and computes a better matching score between language description and the sequential actions (spatial movement + visual observations). Here, we use an explicit memory, as explained in Section 4, to keep track of the spatial movements from a topological perspective.
Table 4: Comparison of our method under different settings: a) three variants of memory: no memory, GPS memory and our memory, and b) two different visual features: ResNet and SphereNet.

| Memory       | Visual Feature | SPL ↑ Nav Err ↓ Ave. Steps ↓ |
|--------------|----------------|-----------------------------|
| No GPS       | ResNet         | 6.87 1374.02 16.52           |
| Ours         | ResNet         | 9.04 742.36 14.60            |
|              | SphereNet      | 10.44 795.90 13.06           |
| Ours         | SphereNet      | 9.37 801.38 13.12            |
|              | Ours           | 11.53 707.13 12.97           |
|              | Ours (Full)    | 11.92 633.85 12.51           |

5.2 Ablation Studies

In order to further understand our method, we perform three ablation studies on memory types, difficulty levels of navigation and visual features.

Memory. We compare our memory to no memory and to a trajectory of GPS coordinates. For GPS coordinates, an encoder LSTM is used to extract features $\psi_3(G_t)$, where $G_t = \{(X_{gps}^i, Y_{gps}^i) : i \in [1, ..., t]\}$. The results are reported in Table 4. We see that Student-Forcing [4] and RPA [67] has a SPL score of 8.77% and 8.43%, respectively (from Table 2). Ours with no memory gets 6.87%. This is because Anderson et al. [4] and Wang et al. [67] use sequence-to-sequence model [57] which has implicit memory about the history of visual path and actions. The poor performance of no memory is also seen in Figure 9 when compared against all other methods either with implicit or explicit memory.

Encoding the explicit memory as a trajectory of GPS coordinates improves SPL from 6.87% to 9.04% as done in Ours (no memory). Ours with GPS memory also performs better than having implicit memory under both SPL and Navigation Error as shown in Table 4. Furthermore, when we employ the top-view trajectory map representation as the explicit memory, our method outperforms all the previous methods either with no memory or with implicit memories as one can see in Table 2 and Table 4. This validates the effectiveness of our explicit memory. It has been found already that top-view trajectory map representation is very useful in learning autonomous driving models [29]. Our findings are in line with theirs, in a relevant but different application.

We take a step further to evaluate the memory types under different evaluation settings of trajectory length and destination threshold. We again compare our method Ours (Full) to Ours (No memory) and Ours (GPS memory). The results are shown in Figure 9. The figure shows that Ours (Full) has higher SPL consistently over others at different trajectory length and destination threshold. Our variant Ours (GPS) where GPS information is encoded as a sequence of coordinates, shows deteriorating performance with increasing trajectory length. Subsequently, this shows the importance of a 2-dimensional representation of the memory as used in Ours (Full). In the case of Navigation error plots, we observe that Ours (Full) performs better than other methods. However, we see that Ours (Full) performs worse than Ours (GPS) at 300% of trajectory length. This may be be-

Fig. 9: Rows for metrics and columns for evaluation settings. Trajectory length: % extra length of path allowed for agents to navigate at evaluation. Destination Threshold (meter): a threshold of the final distance to the destination, below which is considered as a success.

Table 5: Cross-difficulty evaluation of our method. We train our model with training data of routes with each difficulty level {1, 2, 3, 4} and we evaluate for all the models on all difficulty levels. All denotes the mixed routes of all difficulty levels.

| # of landmarks (test) | 1    | 2    | 3    | 4    | All  |
|-----------------------|------|------|------|------|------|
|                       |      |      |      |      |      |
| # of landmarks (train)| 1    | 2    | 3    | 4    | All  |
|                       |      |      |      |      |      |
| 1                     | 76.71| 34.21| 21.41| 8.45 | 45.89|
| 2                     | 75.11| 46.42| 27.07| 10.71| 50.08|
| 3                     | 74.43| 46.62| 28.14| 11.51| 50.41|
| 4                     | 74.28| 43.08| 27.96| 11.92| 49.93|
| All                   | 75.33| 46.10| 27.32| 11.68| 51.37|
Go ahead for two blocks until you reach the highway. You will see a huge electric pole on the right. Turn left and go to the next road junction. There is a two-story building on the left near to the junction. Continue ahead for two blocks. There is a red stoned building on the right. Go straight for two more blocks. You will see a red building on the left. Turn right and go to the next road. Storey building on the right. Huge electric pole turn left. Junction bank with a two-story building on the left near to the junction. Ahead to the next road there is a red stoned building. Go ahead for two blocks. There is a red building on the opposite right road side. You will see some field on both left and right road side. Go straight for two more blocks. You will see an another green bridge on the top. Turn right and go to the next road. Source -> Landmark 1. Landmark 1 -> Landmark 2. Landmark 2 -> Landmark 3. Landmark 3 -> Destination. We did not show all intermediate steps due to space constraints. The right panel shows the final traversed path by the agent along the path embedded with the predicted moving direction, the memory images and the agent's navigated paths. We observe that average steps of Ours (full) increase relatively slower than other methods with the trajectory length. It can be seen that Ours (full) outperforms other methods with the trajectory length and the ground-truth trajectory with different colors indicating different sub-routes between consecutive landmarks. We did not show all intermediate steps due to space constraints. The right panel shows the final traversed path by the agent along the path embedded with the predicted moving direction, the memory images and the agent's navigated paths.
different navigation difficulty levels, we take one step further to study the effect of training and testing our model with routes of variable number of landmarks. We have tabulated in Table 5 the performance of the method under the different combinations of route lengths (i.e. the number of landmarks) used in the training and testing phases.

The table shows that the performance of our model is fairly high when trained with routes of 1 landmark and tested on the same difficulty level. However, the performance drops drastically when tested on routes of higher difficulty levels. The performance on longer routes improves when we train the model with routes of the same or higher difficulty levels. The main conclusion from this experiment is that a model trained with harder cases works well on easier test cases, but not the other way around.

In the last row, we show the model performance when trained with mixed routes of all the difficulty levels. The trained model achieves competitive performance at all navigation difficulty levels and outperforms all other models for the mixed difficulty levels.

One can also see that the performance drops almost exponentially with the level of navigation difficulty. The navigation at difficulty level 4 is already very challenging as the highest SPL is 11.92 only. Hence, annotating even longer routes is not very necessary at the moment for training and validating the current learning algorithms. This experiment is also a showcase of the notable merit of our Talk2Nav dataset that routes of different difficulty levels can be created for the fine-grained evaluation.

**SphereNet vs. ResNet.** We observe that using the pre-trained SphereNet yields better accuracy for the navigation task than training from scratch. This means that our proposed self-learning tasks are useful for model pre-training. However, we see in Table 4 that learning with SphereNet has comparable or slightly worse performance than with ResNet. This may be due to the fact that ResNet is trained on ImageNet which comes with human labels. Henceforth, we evaluate all variants of our models with ResNet. We believe that SphereNet is likely to perform better with architectural changes like residual connections, which can be a promising future work.

5.3 Qualitative Analysis

We also provide qualitative results in a step-by-step fashion in Figure 10. The given input navigational instructions and 360° visual observations at the source node are given on the left of the figure. The rest is devoted to the intermediate and final results during the course of the VLN task. The middle panel of the figure shows the rows of intermediate results: a) the first row shows the results of language segments of landmark descriptions and directional instructions, b) the second row depicts the front 360° views of the agent along the path embedded with action angle prediction in bins, c) the third row shows the memory images written by the write module and, d) the final row shows the agent’s navigated paths predicted by our model. Different stages of finding landmarks are indicated by different colors. The ground truth route with landmarks are on the right of the figure. The numbers on the images denote the number of steps already traversed. We see that when the agent reaches a landmark, the memory is re-initialized and the soft attention over the navigational instruction segments moves forward to the next segment. Due to space constraints, we did not show all the intermediate steps.

In Figure 10, we show two examples of results from our VLN experiments on the Talk2Nav dataset on two different routes. Here, we show a successful navigation in the first example. Though the agent misses the right path at some point, it successfully comes back to the path towards the destination. It takes 14 steps to reach the destination of which only a sub-set of intermediate steps are displayed in the figure. In the second case, we observe that the agent fails to navigate to the destination and forms a loop in the road until it finishes the episode. The failure is partially due to the segmentation error of the language instruction – the overall instruction is segmented into 3 segments instead of 4. This causes confusion for the matching modules. In both the cases, we can clearly see that the memory of the topological view of the path is intuitive to interpret.

6 Conclusion

This work has proposed a new approach for language guided automatic wayfinding in cities using memory framework and soft attention mechanism over language descriptions. The main contribution of our work are: a) an effective method to create large-scale navigational instructions over long-range city environments; b) a new dataset with verbal instructions for 10, 714 navigational routes; c) a novel learning approach of integrating explicit memory framework of remembering the traversed path, and soft attention model over the language segments controlled by Adaptive Computation Time LSTMs. Experiments show that our model outperforms other methods consistently.

References

1. Agrawal, A., Lu, J., Antol, S., Mitchell, M., Zitnick, C.L., Parikh, D., Batra, D.: Vqa: Visual question answering. International Journal of Computer Vision 123(1), 4–31 (2017) 2
2. Anderson, P., Chang, A., Chaplot, D.S., Dosovitskiy, A., Gupta, S., Koltun, V., Kosecka, J., Malik, J., Mottaghi, R., Savva, M., et al.: On evaluation of embodied navigation agents. arXiv:1807.06757 (2018) 11
3. Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., Zhang, L.: Bottom-up and top-down attention for image captioning and visual question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6077–6086 (2018) 3
4. Anderson, P., Wu, Q., Teney, D., Bruce, J., Johnson, M., Sudderth, N., Reid, I., Gould, S., van den Hengel, A.: Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018) 1, 2, 3, 4, 6, 7, 8, 10, 11, 12, 13
5. Andreas, J., Rohrbach, M., Darrell, T., Klein, D.: Learning to compose neural networks for question answering. arXiv preprint arXiv:1601.01705 (2016) 2
6. Aneja, J., Deshpande, A., Schwing, A.G.: Convolutional image captioning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5561–5570 (2018) 3
7. Anne Hendricks, L., Wang, O., Shechtman, E., Sivic, J., Darrell, T., Russell, B.: Localizing moments in video with natural language. In: Proceedings of the IEEE International Conference on Computer Vision (2017) 2
8. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014) 3
9. Balajee Vasudevan, A., Dai, D., Van Gool, L.: Object referring in videos with language and human gaze. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 2
10. Boularias, A., Dulavett, F., Oh, J., Stentz, A.: Grounding spatial relations for outdoor robot navigation. In: IEEE International Conference on Robotics and Automation (ICRA) (2015) 4
11. Brahmabhatt, S., Hays, J.: Deepnav: Learning to navigate large cities. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5193–5202 (2017) 2
12. Chen, D.L., Mooney, R.J.: Learning to interpret natural language navigation instructions from observations. In: Twenty-Fifth AAAI Conference on Artificial Intelligence (2011) 1
13. Chen, H., Shur, A., Misra, D., Snapey, N., Artzi, Y.: Touchdown: Natural language navigation and spatial reasoning in visual street environments. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 2, 3, 4, 6
14. Chopra, S.: Learning a similarity metric discriminatively, with application to face verification. In: IEEE Conference on Computer Vision and Pattern Recognition (2005) 5
15. Coors, B., Paul Condurache, A., Geiger, A.: Spherenet: Learning spherical representations for detection and classification in omnidirectional images. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 518–533 (2018) 11
16. Das, A., Datta, S., Gkioxari, G., Lee, S., Parikh, D., Batra, D.: Embodied Question Answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 1
17. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: IEEE conference on computer vision and pattern recognition (2009) 11
18. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018) 3, 8, 10
19. Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2625–2634 (2015) 3
20. Fried, D., Hu, R., Cirik, V., Rohrbach, A., Andreas, J., Morency, L.P., Berg-Kirkpatrick, T., Saenko, K., Klein, D., Darrell, T.: Speaker-follower models for vision-and-language navigation. In: NIPS (2018) 1, 2, 3, 11, 12
21. Gordon, D., Kemphawi, A., Rastegari, M., Redmon, J., Fox, D., Farhadi, A.: Iqa: Visual question answering in interactive environments. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018) 9
22. Grabler, F., Agrawala, M., Sumner, R.W., Pauly, M.: Automatic generation of tourist maps. In: ACM SIGGRAPH (2008) 3
23. Graves, A.: Adaptive computation time for recurrent neural networks. arXiv preprint arXiv:1603.08983 (2016) 9
24. Graves, A., Wayne, G., Danihelka, I.: Neural turing machines. arXiv:1410.5401 (2014) 3
25. Graves, A., Wayne, G., Reynolds, M., Harley, T., Danihelka, I., Grabska-Barwińska, A., Colmenarejo, S.G., Grefenstette, E., Ramalho, T., Agapiou, J., et al.: Hybrid computing using a neural network with dynamic external memory. Nature 538(7626), 471 (2016) 3, 9
26. Gupta, S., Tolani, V., Davidson, J., Levine, S., Sukthankar, R., Malik, J.: Cognitive mapping and planning for visual navigation. International Journal of Computer Vision (2019) 2
27. Gygli, M., Song, Y., Cao, L.: Video2gif: Automatic generation of animated gifs from video. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016) 5
28. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition (2016) 11
29. Hecker, S., Dai, D., Van Gool, L.: End-to-end learning of driving models with surround-view cameras and route planners. In: European Conference on Computer Vision (ECCV) (2018) 13
30. Hermann, K.M., Hill, F., Green, S., Wang, F., Faulkner, R., Soyer, H., Szepesvari, D., Czarnecki, W., Jaderberg, M., Teyashvin, D., Wainwright, M., Apps, C., Hassabis, D., Blunsom, P.: Grounded language learning in a simulated 3d world. CoRR abs/1706.06551 (2017) 1, 2
31. Hermann, K.M., Malinowski, M., Miorowski, P., Banki-Horvath, A., Anderson, K., Hadsell, R.: Learning To Follow Directions in Street View. arXiv e-prints (2019) 2, 3
32. Hill, F., Hermann, K.M., Blunsom, P., Clark, S.: Understanding Grounded Language Learning Agents. arXiv e-prints (2017) 2
33. Hölscher, C., Tenbrink, T., Wiener, J.M.: Would you follow your own route description? cognitive strategies in urban route planning. Cognition 121(2), 228–247 (2011) 3, 4
34. Hu, R., Andreas, J., Darrell, T., Saenko, K.: Explainable neural computation via stack neural module networks. In: Proceedings of the European conference on computer vision (ECCV) (2018) 3
35. Hu, R., Rohrbach, A., Darrell, T., Saenko, K.: Language-conditioned graph networks for relational reasoning. arXiv preprint arXiv:1905.04405 (2019) 3
36. Hudson, D.A., Manning, C.D.: Compositional attention networks for machine reasoning. arXiv preprint arXiv:1803.03967 (2018) 3
37. Ishikawa, T., Nakamura, U.: Landmark selection in the environment: Relationships with object characteristics and sense of direction. Spatial Cognition & Computation 12(1), 1–22 (2012) 3
38. Karpathy, A., Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions. In: Proceedings of the IEEE conference on computer vision and pattern recognition (2015) 2, 3
39. Ke, L., Li, X., Bisk, Y., Holtzman, A., Gan, Z., Liu, J., Gao, J., Choi, Y., Srinivasa, S.: Tactical rewind: Self-correction via backtracking in vision-and-language navigation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6741–6749 (2019) 2
40. Khosla, A., An An, B., Lim, J.J., Torralba, A.: Looking beyond the visible scene. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3710–3717 (2014) 2
41. Kim, J., Misu, T., Chen, Y.T., Tawari, A., Canny, J.: Grounding human-to-vehicle advice for self-driving vehicles. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 3

Arun Balajee Vasudevan et al.
42. Klippel, A., Tappe, H., Kulik, L., Lee, P.U.: Wayfinding choremesa language for modeling conceptual route knowledge. Journal of Visual Languages and Computing 16(4), 311 – 329 (2005) 4
43. Klippel, A., Winter, S.: Structural salience of landmarks for route directions. In: Spatial Information Theory (2005) 2
44. Kumar, A., Gupta, S., Fouhey, D., Levine, S., Malik, J.: Visual memory for robust path following. In: Advances in Neural Information Processing Systems, pp. 773–782 (2018) 3
45. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11), 2278–2324 (1998) 11
46. Luo, R., Price, B., Cohen, S., Shokminovich, G.: Discriminability objective for training descriptive captions. arXiv preprint arXiv:1803.04376 (2018) 11
47. Ma, C.Y., Lu, J., Wu, Z., AlRegib, G., Kira, Z., Socher, R., Xiong, C.: Self-monitoring navigation agent via auxiliary progress estimation. arXiv preprint arXiv:1901.03035 (2019) 11, 12
48. Ma, C.Y., Wu, Z., AlRegib, G., Xiong, C., Kira, Z.: The regretful agent: Heuristic-aided navigation through progress estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6732–6740 (2019) 1, 2, 12
49. Mansimov, E., Parisotto, E., Ba, J.L., Salakhutdinov, R.: Generating images from captions with attention. arXiv preprint arXiv:1511.02793 (2015) 3
50. Michon, P.E., Denis, M.: When and why are visual landmarks used in giving directions? In: Spatial Information Theory (2001) 2, 3, 4
51. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, pp. 3111–3119 (2013) 10
52. Millonig, A., Schechtner, K.: Developing landmark-based pedestrian-navigation systems. IEEE Transactions on Intelligent Transportation Systems 8(1), 43–49 (2007) 3
53. Mirowski, P., Grimes, M., Malinowski, M., Hermann, K.M., Anderson, K., Teypayshin, D., Simonyan, K., kavukcuoglu, k., Ziserman, A., Hadsell, R.: Learning to navigate in cities without a map. In: NIPS (2018) 2, 4
54. Mirowski, P.W., Pascanu, R., Viola, F., Soyer, H., Ballard, A.J., Banino, A., Denil, M., Goroshin, R., Sifre, L., kavukcuoglu, k., Kumaran, D., Hadsell, R.: Learning to navigate in complex environments. In: ICLR (2017) 2
55. Nguyen, K., Dey, D., Brockett, C., Dolan, B.: Vision-based navigation with language-based assistance via imitation learning with indirect intervention. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 2
56. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for distributed word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532–1543 (2014) 10
57. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in neural information processing systems, pp. 3104–3112 (2014) 13
58. Thoma, J., Paudel, D.P., Chhatkuli, A., Probst, T., Gool, L.V.: Mapping, localization and path planning for image-based navigation using visual features and map. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 2
59. Tom, A., Denis, M.: Referring to landmark or street information in route directions: What difference does it make? In: International Conference on Spatial Information Theory, pp. 362–374. Springer (2003) 4
60. Tom, A., Denis, M.: Language and spatial cognition: Comparing the roles of landmarks and street names in route instructions. Applied Cognitive Psychology: The Official Journal of the Society for Applied Research in Memory and Cognition 18(9), 1213–1230 (2004) 3
61. Tversky, B., Lee, P.U.: Pictorial and verbal tools for conveying routes. In: C. Freksa, D.M. Mark (eds.) Spatial Information Theory. Cognitive and Computational Foundations of Geographic Information Science, pp. 51–64 (1999) 2, 3, 4
62. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017) 3
63. Vogel, A., Jurafsky, D.: Learning to follow navigational directions. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 806–814. Association for Computational Linguistics (2010) 4
64. de Vries, H., Shuster, K., Batra, D., Parikh, D., Weston, J., Kiela, D.: Talk the walk: Navigating new york city through grounded dialogue. arXiv preprint arXiv:1807.03367 (2018) 6
65. Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., Wang, X., Tang, X.: Residual attention network for image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3156–3164 (2017) 3
66. Wang, X., Huang, Q., Celikyilmaz, A., Gao, J., Shen, D., Wang, Y.F., Yang Wang, W., Zhang, L.: 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, pp. 6629–6638 (2019) 2, 3
67. Wang, X., Xiong, W., Wang, H., Yang Wang, W.: Look before you leap: Bridging model-free and model-based reinforcement learning for planned-ahead vision-and-language navigation. In: ECCV (2018) 1, 2, 3, 11, 12, 13
68. Weissenberg, J., Gygli, M., Riemenschneider, H., Van Gool, L.: Navigation using special buildings as signposts. In: Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Interacting with Maps, pp. 8–14. ACM (2014) 3
69. Weyand, T., Kostrikov, I., Philbin, J.: Planet - photo geolocation with convolutional neural networks. In: European Conference on Computer Vision (ECCV) (2016) 2
70. Wortsman, M., Ehsani, K., Rastegari, M., Farhadi, A., Mottaghi, R.: Learning to learn how to learn: Self-adaptive visual navigation using meta-learning. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 2
71. Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al.: Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144 (2016) 2
72. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. In: International conference on machine learning, pp. 2048–2057 (2015) 3
73. Yang, Z., He, X., Gao, J., Deng, L., Smola, A.: Stacked attention networks for image question answering. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 21–29 (2016) 3
74. Zang, X., Pokle, A., Vázquez, M., Chen, K., Niebles, J.C., Soto, A., Savarese, S.: Translating navigation instructions in natural language to a high-level plan for behavioral robot navigation. CoRR abs/1810.00663 (2018). URL http://arxiv.org/abs/1810.00663 3, 6
75. Zhu, X., Li, L., Liu, J., Peng, H., Niu, X.: Captioning transformer with stacked attention modules. Applied Sciences 8(5), 739 (2018) 11
76. Zhu, Y., Kiros, R., Zemel, R., Salakhudinov, R., Urtasun, R., Torralba, A., Fidler, S.: Aligning books and movies: Towards story-like visual explanations by watching movies and reading books.
In: Proceedings of the IEEE international conference on computer vision, pp. 19–27 (2015) 10

78. Zhu, Y., Mottaghi, R., Kolve, E., Lim, J.J., Gupta, A., Fei-Fei, L., Farhadi, A.: Target-driven Visual Navigation in Indoor Scenes using Deep Reinforcement Learning. In: ICRA (2017) 2